

IN-BI-CR
890
169P
431073

SSD92D0154

AUTOMATION LIFE-CYCLE
COST MODEL

FINAL TECHNICAL REPORT

JULY 2, 1992

CONTRACT NAS2-13419
PARAGRAPH F.4(c)

INFORMATION SCIENCES LIBRARY
AMES RESEARCH CENTER
MOFFETT FIELD, CALIF.

JUL 21 1992



Rockwell International
Space Systems Division
12214 Lakewood Boulevard
Downey, CA 90241

N94-27866

Unclas

G3/81 0000890

(NASA-CR-195695) AUTOMATION
LIFE-CYCLE COST MODEL Final
Technical Report (Rockwell
International Corp.) 169 p

← R

1000 1000 1000 1000

1000

1000

1000

Table of Contents

Prologue.....	1
1. Introduction.....	1
2. Problem Statement.....	2
2.1 Designing The Optimal Mix Of Automated And Manually Operated Systems.....	2
2.2 Limitations of Cost Estimating Tools.....	4
2.3 Decision Analysis.....	6
2.4 Problem Summary.....	7
3. Objectives.....	7
3.1 Far-term Objectives.....	8
3.1.1 General Requirements.....	8
3.1.1.1 General Capabilities.....	9
3.1.1.2 Comparative Analysis.....	10
3.1.1.3 Output/Display Formats.....	10
3.2 Near-term Objectives.....	11
4. Approach.....	12
4.1 Overview.....	12
4.2 Existing Model Identification.....	14
4.3 Existing Data Source Identification.....	18
4.4 Advanced Technology Impact Determination.....	19
4.4.1 Assessment of Technology Drivers and Technology Readiness Levels for Robotic Systems.....	20
4.4.1.1 Mobile Base Platform.....	20
4.4.1.2 Manipulator Subsystems.....	21
4.4.1.3 Special Tools/End-effectors.....	23
4.4.1.4 Rockwell Space Systems Division Involvement in Robotics and Related Technologies.....	24
4.4.1.5 Identification of Existing LCC Methodologies.....	24
4.5 Decision Modelling System (DEMOS) Tool Capabilities.....	25
5. Automation Life-cycle Cost Model (ALCM).....	26
5.1 Overview.....	26
5.2 System Architecture Definition.....	28
5.2.1 Generic Life-cycle Cost Categories.....	29
5.2.1.1 Research and Development (R&D) Phase Cost Categories.....	30
5.2.1.2 Investment and Acquisition (I&A) Phase Cost Categories.....	31
5.2.1.3 Operation and Support (O&S) Phase Cost Categories.....	32
5.2.2 Specific Cost Categories.....	33
5.3 Model Development.....	33
5.3.1 Background.....	34
5.3.2 Approach.....	35
5.3.3 Development of Cost Estimating Rationale For Robotic Systems/Subsystems.....	38
5.4. Framework Prototype.....	42
5.4.1 Framework Overview.....	42
5.4.2 Major Components.....	46
5.4.2.1 Phase Definitions.....	47

5.4.2.2	Rate Breakdown.....	50
5.4.2.3	Program-specific WBSs.....	51
5.4.2.4	Cost Analysis Features	55
5.4.2.5	Automation Bottom-line Analysis.....	55
5.4.2.6	Software Cost Modelling	56
5.4.2.6.1	Autotrade Overview.....	57
5.4.2.6.2	COCOMO Cost Drivers.....	59
5.4.2.6.3	Size and Productivity	59
5.4.2.6.4	Autonomy Levels Definition	61
5.4.2.6.5	CSCI Identification.....	62
5.4.2.6.6	Summary Analysis.....	64
5.4.2.6.7	Autotrade Summary.....	65
5.4.3	Framework Capabilities Summary.....	65
6.	Model Validation.....	66
6.1	Overview	66
6.2	Sample Scenario.....	67
7.	Comments and Conclusions.....	67
7.1	Suggested Improvements to the ALCM.....	69
7.2	Follow-on ALCM Effort Statement of Work	72
8.	References.....	78
9.	Appendices	79
9.1	Technical Interchange Meeting (TIM) #1 Notes	
9.2	Technical Interchange Meeting (TIM) #2 Notes	
9.3	ALCM Tool User Notes	
9.4	Programmatics	
9.5	Qualitative Criteria for Robotics CER Generation	
9.6	Robotics DDT&E CER Tables	

List of Figures

Figure 1. The ALCM Tool is the Result of Cooperative Determination of Envisioned Capabilities Between NASA and Rockwell.	7
Figure 2. Near-term Objectives are an Achievable Subset of the Envisioned Capabilities of the Final ALCM Tool.	12
Figure 3. Program Life-cycle Phases and Functions for Space Systems Define the Modules for the ALCM Tool.	14
Figure 4. Life-cycle Cost Methodology Flow Interfaces With All Aspects of a Program.	26
Figure 5. ALCM Life-cycle Phase Definitions.	29
Figure 6. Sample ALCM Work Breakdown by Life-cycle Cost Phase.	33
Figure 7. The ALCM Approach Capitalizes Upon a Large, Known Body of Pertinent Data, Advanced Robotic Applications, and Standardized Costing Methods to Address the Entire Life-cycle Cost for Space Systems.	36
Figure 8. Reach-related Costs Rise Quickly Over 40 Feet.	41
Figure 9. Payload Capability Cost is Very Dependent Upon Weight Until Around 200 Pounds.	41
Figure 10. Cost Associated with Positioning Accuracy Rises Dramatically with Requirements Less than 0.1 Inch.	42
Figure 11. Near-term Objectives are an Achievable Subset of the Envisioned Capabilities of the Final ALCM Tool.	43
Figure 12. Influence Diagrams Are Used to Construct Sophisticated Models in a Readily Understandable Format.	45
Figure 13. The Top Level of the Prototype Contains the Major Model Components for Ease of Use and Navigation.	47
Figure 14. The Research and Development Cost Categories are Implemented as a List Which is Capable of Being Edited.	48
Figure 15. Investment and Acquisition Costs Include Non-recurring Elements Which Can Be Tailored Per Project.	49
Figure 16. Operation and Support Costs Typically Account for the Bulk of Life-cycle Cost and Can Be Reviewed in Detail.	50
Figure 17. Time-dependent Factors of Cost are Modelled Through the Rate Breakdown Submodel.	51
Figure 18. The ALCM Framework Supports Multiple Cost Estimation Methodologies and Their Comparative Analysis.	52
Figure 19. Different Estimation Methodologies Are Applied to Appropriate Costing Situations.	53
Figure 20. The ALCM Framework Prototype Provides a Flexible Environment for Constructing and Reviewing Space System Cost Models Based on Work Breakdown Structures.	54
Figure 21. ALCM Framework Provides Cost Analysis Capabilities to Review Cost by Phase and Element to Determine Key Drivers in the Life-cycle.	55
Figure 22. The Bottom-line Figure is Available From the Model Without Examining an Immense Amount of Detail.	56
Figure 23. Autotrade is Functionally Partitioned to Ease Navigation of the Model.	58



THE UNIVERSITY OF CHICAGO PRESS



Figure 24. Autotrade Automatically Assigns the Correct Numeric Value to Each COCOMO Cost Driver to Provide Consistency.....	59
Figure 25. Autotrade Permits Uncertain Size Estimates for Each CSCI to be Used for Derivation of Productivity Parameters.....	60
Figure 26. CSCI Productivity Measures May Be Analyzed From Several Perspectives.	61
Figure 27. Specification of Autonomy Level per CSCI Supports Software Cost/Benefit Analysis for Space Systems.....	62
Figure 28. Cost Driver Values Are Specified For Each CSCI Permitting Detailed Analysis of the Envisioned System.....	63
Figure 29. COCOMO Cost Factors May be Examined With Their Assigned Values per Autonomy Level and In Rank Order to Obtain Their Relative Importance.....	64
Figure 30. Summary Trade-off Analysis is Supported by Presenting Costs and Productivity Measures Over a Range of Autonomy Levels.....	65
Figure 31. Sample ALCM Object Definition Window With Provisions For System Descriptions and Data Legacy/Fidelity.	70
Figure 32. Parametric vs. Accounting Methods For Determining TFU Costs For The Various Elements of A Sample Robotic System.....	71
Figure 33. ALCM Model Will Perform Simultaneous Bottom-Up and Top-Down Cost Analysis and Assess Validity of Parametric Relationships.	72



List of Tables

Table 1. Summary of ATAC Automation and Robotics Findings.....	1
Table 2. A&R Applications Can Especially Benefit the Operation and Maintenance of a Space System.	2
Table 3. Previous Productivity Improvement Results Have Been Qualitative.	3
Table 4. Problems with Traditional Software Tools for Modelling.....	6
Table 5. Far Term ALCM Capabilities Will Support Management Decisions And Ensure Low LCC With Identified Uncertainty.....	8
Table 6. The Approach Ensures an Interim Product to Assist Tool Acceptance and Validity.....	13
Table 7. Utilizing Three Different Analysis Perspectives Provides the Opportunity for a More Robust and Useful ALCM Tool.	13
Table 8. Existing Models Are Leveraged to Accelerate the Pace of ALCM Tool Demonstration and Development.....	16
Table 9. Use of NASA Technology Readiness Levels Can Facilitate Risk and Development Cost Generation.....	17
Table 10. Summary Findings of Data Sources Reviewed.	19
Table 11. DEMOS is a Highly Flexible and Appropriate Environment for the Solution of LCC Problems.	25
Table 12. DEMOS Features Can Be Easily Tailored for the ALCM Tool.	25
Table 13. Research and Development Phase Cost Category Definitions.	30
Table 14. Investment and Acquisition Phase Cost Category Definitions.	31
Table 15. Operation and Support Cost Category Phase Definitions.....	32
Table 16. Spreadsheet Analysis of Mobile Base Platforms To Determine Calibration Coefficients.....	38
Table 17. CER Generation is Highly Reliant Upon Experienced Personnel and Mathematical Expression Manipulation Tools.	39
Table 18. Spreadsheet Analysis of Manipulator Systems To Determine Calibration Coefficients.....	40
Table 19. The Manipulator System CER Uses all Associated Major Design Parameters.....	40
Table 20. The ALCM Framework Prototype Has Met the Program's Objectives.	66
Table 21. The ALCM Framework Examines the Entire System Life-cycle to Maximize Cost Savings Potential.	68



List of Acronyms

A&R	Automation and Robotics
AIAA	American Institute of Aeronautics and Astronautics
ALCM	Automation Life-cycle Cost Model
ATAC	Advanced Technology Advisory Committee
CER	Cost Estimating Relationship
COCOMO	Constructive Cost Model
CSCI	Computer Software Configuration Item
DDT&E	Design, Development, Test, and Evaluation
DEMOS	Decision Modelling System
DoD	Department of Defense
DOF	degree of freedom
DTLCC	Design To Life-Cycle Cost
GE	General Electric
GUI	Graphical User Interface
HART	Humans/Automation/Robotics/Telerobotics
I&A	Investment and Acquisition
IOC	Initial Operating (Operational) Capability
IR&D	Independent Research and Development
LCC	Life-Cycle Cost
LRU	Line Replaceable Unit
MMI	Man/machine interface
NASA	National Aeronautics and Space Administration
O&S	Operations and Support
OPF	Orbiter Processing Facility
ORLA	Optimum Repair Level Analysis
PEP	Producibility Engineering and Planning
PLANET	Planetary Logistics Analysis and Evaluation Tool
PLS	Personnel Launch System
PRAF	Production Rate Adjustment Factors
R&D	Research and Development
RPAL	Rockwell Palo Alto Laboratory
RSOC	Rockwell Shuttle Operations Company
SEI	Space Exploration Initiative
SLOC	Source Line of Code
SOW	Statement of Work
SSD	Space Systems Division (Rockwell International)
TFU	Theoretical First Unit
TIM	Technical Interchange Meeting
U.S.	United States
WBS	Work Breakdown Structure

Prologue

This report is submitted to satisfy the contractual requirement of a final technical report. Its purpose is to provide concrete evidence of proper progress on this contract. All information, data, and models created during this effort are described in this document. This document is structured to coincide with the Statement of Work (SOW) for this contract. Section 7 (Comments and Conclusions) consists of an overview of the satisfaction of the SOW tasks. Additionally, recommendations for the future direction and capabilities of the Automation Life-cycle Cost Model (ALCM) are noted. Appendices include additional information on Technical Interchange Meeting (TIM) charts, programmatic information, and Decision Modelling System (DEMOS) reference material.

1. Introduction

The United States Congress mandated, in 1984, that the National Aeronautics and Space Administration (NASA) vigorously advance the arts of automation and robotics (A&R) as they apply to space programs. In addition, the NASA Administrator was tasked to establish an Advanced Technology Advisory Committee (ATAC) in conjunction with NASA's Space Station program. Among the activities performed by the ATAC, was the generation of a report that was intended to provide top level guidelines and requirements for effective implementation of A&R into space systems. Table 1 presents a summary of the preliminary ATAC findings.[2]

- Criteria for the incorporation of A&R technology should be developed and promulgated.
- Verification of the performance of automated equipment should be stressed, including terrestrial and space demonstrations to validate technology for Space Station use.
- Maximum use should be made of technology developed for industry and government.
- The techniques of automation should be used to enhance NASA's management capability.
- An evolutionary station should achieve, in stages, a very high level of advanced automation.
- An aggressive program of long-range technology advancement should be pursued, recognizing areas in which NASA must lead, provide leverage for, or exploit developments.
- A vigorous program of technology transfer to U.S. industries and research communities should be pursued.
- NASA should provide the measures and assessments to verify the inclusion of A&R in the Space Station

Table 1. Summary of ATAC Automation and Robotics Findings.

A recurring theme throughout the ATAC findings was the requirement for identification and development of management control parameters, corresponding to A&R design/development and implementation. The final item listed in Table 1 states a requirement to quantitatively trade-off various configurations using variations in the use of A&R in space systems. More specifically, measures of effectiveness criteria, assessment aids, and utility models were identified as methods to support management objectives and to accurately project future system capabilities and costs. The ATAC explicitly recommended

the development of life-cycle cost (LCC) models and methods for evaluating the Space Station program, as a method for projecting costs throughout the life of a program, and to ease management decisions and planning.

In the context of this contract, the term automation applies to systems ranging from physical devices that mechanize a particular process to the application of knowledge-based systems. Robotic devices would range from tele-operated machines to fully autonomous vehicles, deployed either in space or on the ground (distant planet). While A&R definitions are germane to the development of an automation cost estimation/analysis type model (to the degree that an LCC model would have to be sensitive to varying degrees of A&R technology), emphasis is not directed towards analysis of A&R technical properties, rather to how the varying degrees/levels effect overall program costs.

2. Problem Statement

Two analyses instrumental to the selection of automated versus manually operated systems are: (1) LCC analysis, and (2) comparative analysis. The results of these two types of analyses are subject to the limitations of the methods beings applied, and it is typically difficult to assess the accuracy/quality of the input or output data.

2.1 Designing The Optimal Mix Of Automated And Manually Operated Systems

A key issue in planning future space projects is deciding the appropriate level of A&R. In a specific program, such as the Space Exploration Initiative (SEI), we must decide which tasks are better performed by astronauts, and which by A&R. Table 2 presents a list of the issues and/or trades that must be addressed.[2] As A&R technology advances, how and when should those advances be applied to operational systems? The following parameters are critical: cost, technology maturity, complexity, maintainability, and reliability. Employing automation is not an either/or question, but one of degree. The question is what level of automation and autonomy will provide the optimal trade-off between performance and cost.

- Life-cycle operations costs will overwhelm start-up costs. Later operational costs could restrict or terminate the program.
- Costs of initial A&R hooks and scars affect life-cycle-cost (LCC) savings.
- Automated systems may lower probability and costs of accidents.
- The level of A&R leverages the value of astronaut time.
- A&R can improve support and increase crew time for user payload.
- Extra ground support may be needed for A&R technology and space robotics.
- Space testing of A&R systems introduces new costs.
- Extensive ground-based control may overload communications capacity.
- Faster response time resulting from on-board data analysis will lower costs. (This can be substantial if an on-board expert system does trend analysis.)
- Use of tele-operated, instead of autonomous, systems will raise cost of training astronauts.

Table 2. A&R Applications Can Especially Benefit the Operation and Maintenance of a Space System.

The impact of the degree of A&R and human participation on a given program is manifested during program operational phases, while the decision who performs what tasks, and how much automation is incorporated into the system are all made during the design and development phases. There are myriad types of analyses that could be, and are, performed to assist program managers in choosing between A&R and manual systems (Table 3 summarizes an example.) Yet, most of the trades and issues can be resolved by performing two very common types of analyses: cost analyses and task analyses. An LCC model contains nearly all the elements of a task analysis type model: the primary difference being that a task analysis, along with addressing cost issues, also addresses system availability and maintenance concerns.

System Type	Workforce *	Speed	Accuracy	Training Time	Consistency
Monitoring & Control	●	●	◐	◐	●
FDIR (Fault Detection, Isolation, and Recovery; used here to include Fault Diagnosis)	◐	●	◐	●	●
Planning & Scheduling (including dynamic replanning/rescheduling)	●	●	◐	◐	●
Human Computer Interface	◐	●	●	●	●
Training	◐	◐	○	●	●

Note: Workforce savings may be in number of people or decrease in required work.

LEGEND

○ no improvement

◐ moderate improvement

● significant improvement

Source: Space Station Freedom Automation and Robotics: An Assessment of the Potential for Increased Productivity, March 1990.

Table 3. Previous Productivity Improvement Results Have Been Qualitative.

National interest and emphasis are oriented toward development and deployment of cost effective systems. Program and system decisions should be made which support the most economical options with respect to innate A&R and human capabilities/limitations. Other engineering tradeoffs will be performed to down-select the optimal manual and automated systems, and comparative cost analysis will ensure that the most economical alternatives emerge as the leading system candidates.

2.2 Limitations of Cost Estimating Tools

Developing cost estimates using automated or even manual methods and techniques is not new. In fact, there are just as many, if not more, models/tools as there are operational systems. There are potentially many reasons for this, however, one can intuitively conclude that new models and tools arise because the previously existing capabilities contained some form of inadequacy. These inadequacies include:

- (1) The assumptions, structure, and limitations of cost models are not clearly documented.
- (2) The source, quality, uncertainty, and relevance of input data are not clearly documented.
- (3) The limitation and uncertainties in the output results are not clearly documented.
- (4) Cost models have been built to estimate the cost of a design after the design has been specified, rather than developed as an integral part of the design process to help with early design decisions.
- (5) Separate cost models are built for different project phases and subsystems, often using different assumptions, making it difficult to combine them to assess the full life-cycle cost for the system.
- (6) Cost models use only a single methodology, and do not support multiple approaches, such as top-down, parametric cost estimation, and bottom-up methods for cross-validation.

Customers and contractors alike struggle with the limitations of current cost estimating tools. This is due primarily to the fact that a model's construction, assumptions and limitations are typically not well documented. Cost estimation models that address, equally, all components and phases of a program are rare. Therefore, the results will reflect the limitations of the model/tool. There are three principal reasons why cost estimating tools do not represent a "true" reflection of system life-cycle costs.

- (1) The nature of a model may not match the particular phase of a given program. For instance, accounting-type models (i.e., PRICE) require detailed information to project system LCC, and their application during conceptual phases of a system would tend to skew results. Similarly, it would not be appropriate to employ parametric estimating tools or techniques to mature programs where sufficient data exists to populate an accounting model to support management assessment and projection requirements.

-
- (2) Models that are "generic" in nature may not be suited to perform assessments on specific and complex systems/projects.
 - (3) Models tend to reflect the experience of their designers: models developed by firms very knowledgeable about system operational phases will tend to well represent recurring cost categories and poorly treat the non-recurring cost categories; and those familiar with design, development and production will tend to embellish models with those features, while treating recurring operations and support costs as percentages of the non-recurring cost categories.

The accuracy/validity of any model's output is directly related to the quality or accuracy of the input data. There has been, in the past, no method inherent in existing cost estimating tools, to quantitatively assess the relative significance of the input data. Understanding the input data is necessary to ascertain its relevancy to the project under consideration. For instance, estimated unit production cost may be based on prior experience related to similar units. But how similar is the proposed unit to the experience-based unit: Is it constructed of the same material? Can the same manufacturing process be used? Is there a difference in the manufacturing technology? Is the estimate entered into the model as a vendor quote or an engineering estimate? and do they include fees? This simple example illustrates the risk associated with the quality and relevance of input data on the accuracy of the resultant model output.

When considering the limitations of a particular model and its input data, one tends to view the results with suspicion or outright skepticism. The situation is exacerbated when a customer cannot duplicate and therefore validate the data. If some form of risk assessment does accompany cost data, it is usually at the cost of a separate and rigorous modeling exercise, one that is fraught with its own assumptions and limitations.

Finally, life-cycle cost assessments have been traditionally limited to projecting program costs after the designers have selected a design concept (usually based on performance merits). This is a downfall in the model construction, the program organizational structure, and the product development process. First, most cost estimating models are constructed in such a manner that all input data is gathered, once the design functions have finished an iteration of the design. The data is run through the model and discrete values are then provided to management/ customer. Further, typical models do not generally provide the granularity to determine exactly which elements of a system are driving non-recurring, recurring, or total LCC. To achieve an optimally low cost system, designers must be able to evaluate the cost (nonrecurring, recurring, and total life-cycle) of their alternative designs while they are still on the drawing boards. Models are currently not constructed to support rapid and user friendly component trade studies and sensitivity analyses. Second, the product development process and associated organization are not generally structured to facilitate design to LCC—again cost estimating is something done after the system and its components are designed.

2.3 Decision Analysis

Implicit in the problems listed above, is the accuracy of the required model inputs. Projecting future costs, and future capabilities, is fraught with peril. A method must be encapsulated in the model to provide realistic assessment of LCC outcomes. This method should allow for alternative options, explicit representation of system states and relationships, provide a means for model evolution, and account for the uncertainty of the problem domain.

Real-world problems inherently contain partial data. Extrapolation and abstraction methods may be applied to remedy this situation so that the domains might be modelled. Modelling by its very nature requires abstraction of portions of the problem domain so that there is a tractable solution. Traditional modelling techniques typically embrace these abstractions wholeheartedly and weave them into the solution space. A more correct method is to explicitly account for those abstractions and model them directly. In this manner, confidence in the solution and insight into the problem domain are expanded.

Recent progress in decision analysis theory and tools have made it practical for their application to real-world problems. "Probability and decision theory provide a set of principles for rational inference and decision making under uncertainty...Decision analysis is the art and science of applying these ideas to provide practical help for decision making in the real world." [3] Decision theory permits the explicit statement of preferences among alternate possibilities with associated levels of belief and resolves their combination within a consistent computational domain. Applying this theory permits sensitivity analyses to be conducted on a problem space so that key drivers to the ultimate solution may be readily identified. This capability permits the focusing of energy on those important parameters within the problem so that the uncertainty might be reduced. Hence, confidence increases in the ultimate solution of the problem model.

Modelling cost, with its tools and environment, is typically a stumbling point. The purpose of generating of a model initially is not only to definitively describe a solution to a problem, but also the development of a consistent and timely solution. As such, software models are commonplace to address these issues. However, the solution environment of software brings its own set of constraints to the modelling domain. Table 4 summarizes typical problems with traditional software modelling tools.

- | |
|---|
| <ul style="list-style-type: none">• Poor visualization of model structure• Inadequate treatment of uncertainty• Poor documentation of assumptions• Cumbersome to review, revise, or extend• Insufficient iterative refinement• Inappropriate detail• Results not trusted by decision makers |
|---|

Table 4. Problems with Traditional Software Tools for Modelling.

2.4 Problem Summary

On the basis of the direction provided by the U.S. Congress, coupled with the increasing maturity of A&R technology, space system development efforts are actively pursuing the use of automation techniques. Risk and task analyses are being performed to estimate the impact of including this technology. Management requires effective and timely projections by management to determine the viability of A&R implementation in the various programs. One means of justification is the use of LCC models to support trade-off analyses for profitability. The leap technology proposed warrants, if not requires, extrapolation methods and explicit handling of uncertainty. Software models are increasingly used to generated high fidelity projections of system performance. These models, however, have historically compiled assumptions and judgments into the model giving the end user little insight into the reasoning process. Confidence in the model results is reduced and is typically viewed with skepticism.

The problem domain being addressed by this contractual effort is summarized by the following list:

- A&R technologies appear to viable alternatives to current, manual operations
- life-cycle cost models are typically judged with suspicion due to implicit assumptions and little associated documentation
- uncertainty is a reality for increasingly complex problems and few models explicitly account for its affect on the solution space

3. Objectives

Two objectives were set forth at the outset of this study effort. The first objective is to identify and describe far term capabilities and requirements envisioned for the Automation Life-cycle Cost Model. Following definition of model requirements, the second objective establishes a framework development approach that supports achievement of the defined requirements. Figure 1 is graphical depiction of the relationship among the objectives and the ultimate ALCM tool.

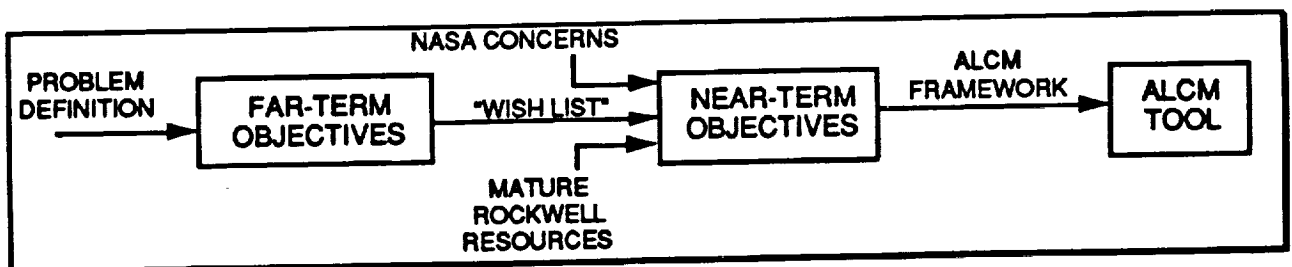


Figure 1. The ALCM Tool is the Result of Cooperative Determination of Envisioned Capabilities Between NASA and Rockwell.

The far-term objectives document the envisioned capability within 3-5 years of effort. These objectives are consistent with the "ultimate" tool proposed by LCC analysts. This vision captures capabilities and shortcomings of existing tools. Near-term objectives are a little more "down to earth." Given the NASA concerns for LCC values and mature Rockwell resources, a compromise is reached on satisfying the "wish list" of far-term objectives. The near-term objectives are capable of being included in an ALCM tool within 1-2 years of contractual effort. The overall goal of this contractual effort is the definition of these objectives and the generation of a software framework demonstrating the near-term objective capabilities.

3.1. Far-term Objectives

The capabilities of the ALCM will coincide with the issues outlined in Section 2.0 (Problem Statement), specifically: LCC analysis, task analysis and uncertainty analysis. It is intended that the ALCM should not simply be a high fidelity LCC tool, but one that assists in the design decisions and can give a manager the kind of information necessary to support programmatic and system decisions. Table 5 depicts the capabilities currently envisioned for the ALCM.

General Capabilities	Sensitivity/Tradeoff Analysis Capabilities	Output/Display Formats
• Assessment/Determination Tools	• Temporal (e.g., duration, frequency)	• Spreadsheet Paradigm
• Risk Analysis	• Quantity	• Graphical
• Temporal Representation	• Sparing	• Model Structure
• Integration with Existing Databases	• Manual vs. Automation	
• Security of Data/Algorithms	• Level of Automation	
• Integrated Documentation		
• Multiple Fidelity Levels		

Table 5. Far Term ALCM Capabilities Will Support Management Decisions And Ensure Low LCC With Identified Uncertainty.

3.1.1 General Requirements

Incorporation of the capabilities/requirements depicted in Table 5 represent: (1) the typical variables which would effect the outcome of an A&R versus manual trade study, (2) variables associated with what-if analyses the customer or program manager are likely to desire/request, (3) a method for rapidly and easily performing comparative tradeoff analyses, and (4) alternative methods to view data, that would present different perspectives of data. The model should be capable of easily and rapidly supporting comparative analysis and depicting results in an easily interpretable graphic or spreadsheet format. Any requirement that may arise to use either pre- or post-processors, or any other external module, in association with the main model/environment shall be transparent to the user.

3.1.1.1 General Capabilities

The General Capabilities category of objectives includes those items that should be present in the ALCM tool in its envisioned configuration to assist in the overall operability of the package. These requirements reflect the operational modes in which the tool could be used. The following list provides a more detailed explanation for each of the parameters:

- **Assessment/Determination Tools**—These capabilities provide a cost analyst the ability to derive costing relationships and drivers for particular applications. Items may include design, operations, and maintenance cost drivers.
- **Risk Analysis**—These capabilities provide the ability to specify and model parameters in the problem space that may contribute to the risk of the solution. Decision analysis tools support the uncertain portions in the above specifications. Uncertain parameters may include cost, technology maturity, and sizing values.
- **Temporal Representation**—The framework provides the capability to explicitly model time-variant values and relationships. For instance, the tracking of cost elements over time is required.
- **Integration with Existing Databases**—The ability to access and utilize existing sources of pertinent data greatly enhances tool confidence and reduces development risk. By capitalizing upon known quantities, the ALCM tool becomes a new interface to pre-existing models. This capability supports the other facets in this category by providing the raw data for analysis. The capability shall also be provided to enter new, and revise existing, data within the tool environment.
- **Security of Data/Algorithms**—The exclusion of particular portions of the ALCM tool environment is required to protect sensitive and proprietary data from unauthorized access. The separation of usable and partitioned data under tool control facilitates configuration management.
- **Integrated Documentation**—The incremental development and evolution of a model should be reflected in its documentation. The integration of documentation capabilities with the modelling parameters themselves encourages the updating of material. Explanation, legacy, and definitive information may be included.
- **Multiple Fidelity Levels**—The ability to represent various levels of fidelity promotes prototyping of systems and accelerated definition of the problem domain. This abstraction mechanism allows the user to concentrate on a localized aspect without the burden on the overall system. Multiple levels promote understandability by the end user through the reduction of clutter in model presentation.

3.1.1.2 Comparative Analysis

The Comparative Analysis capabilities of the envisioned ALCM tool provide the means of performing sensitivity and trade-off analyses. The exploration of alternative possibilities within the context of the tool environment is a powerful analysis ability. The rapid generation and examination of pertinent data displayed regarding other possible situations promotes the discarding of irrelevant or risky solutions. The testing of more situations also enhances the confidence in the ultimate decision supported by the ALCM tool. Interpretation and experimentation within the confines of the tool environment reduce the chances of translation and transcription errors. The following list provides a more detailed explanation for each of the parameters listed in this category in Table 5.

- **Temporal**—LCC inherently includes time-variant relationships and data. The associated analysis, likewise, includes the viewing of parameters as they change with time. Examination of those changes (or lack of change) may indicate significant performance of that system that ultimately affects the total system cost. Examples of temporal representations include life-cycle duration, mission frequency, mission duration, and quiescent periods.
- **Quantity**—The number of systems (and their constituent parts) and the extent of the missions involved heavily influence life-cycle costs. For example, the explicit representation of these parameters is useful for determining breakeven points, and optimal production values and rates. Values may include number of operating systems and component quantities.
- **Sparing**—Related to the Quantity parameter, the number of spares of a unit can heavily influence LCC. This value is dependent on the use of manual or automated means. The evaluation of this highly dependent parameter is useful throughout all phases of the life-cycle.
- **Manual vs. Automation**—One of the main goals of this contractual effort is the generation of evidence to support trade-off analyses of the use of A&R. The ability to specify the use of automation at various levels of system design is an important feature of the ALCM tool.
- **Level of Automation**—Automation is typically not a discrete function when implemented. Varying levels of automation are desirable to maximize system utility and minimize LCC. The specification and analysis of this parameter will heavily influence system design and LCC.

3.1.1.3 Output/Display Formats

The Output/Display Format capabilities need to maintain consistency with traditional mechanisms of LCC analysts and provide new insight into the advanced capabilities of the ALCM tool. Legacy from models being implemented in database and spreadsheet tools is apparent from desired model outputs being tables and simple graphs. The addition to this

suite of formats is a mechanism for display, and to assist manipulation, of the model structure itself. The following list provides a more detailed explanation for each of the parameters:

- **Spreadsheet Paradigm**—Tables depicting the relationship of two different variable categories are a preferred method of viewing LCC data. This capability is useful and provides a familiar interface to the end user, promoting acceptance of the ALCM tool.
- **Graphical**—Simple graphs of the variable categories from various perspectives are also a preferred method of viewing LCC data. This capability is associated with spreadsheet paradigm. The ability to show two or more variable categories at one time is a useful capability of the ALCM tool. Types of graphs include histogram, pie, 2D/3D variations, and line (variable values and uncertainty distributions).
- **Model Structure**—The ALCM tool models are envisioned as being highly modular and hierarchical in nature. Standard diagramming paradigms for model construction and display will be used. The hierarchy will assist in organizing large models and promote abstraction for understandability. The standard diagramming techniques will enforce model construction consistency and understandability. The techniques will have an associated graphical representation that is both expressive and easy to use.

3.2. Near-term Objectives

As previously defined, the Near-term Objectives are a subset of the Far-term Objectives. Near-term relates to 1-2 years of contractual effort resulting in a product that supports the identified objectives. Figure 2 provides a graphical representation of the Near-term Objectives within the context of the overall vision for the ALCM tool.

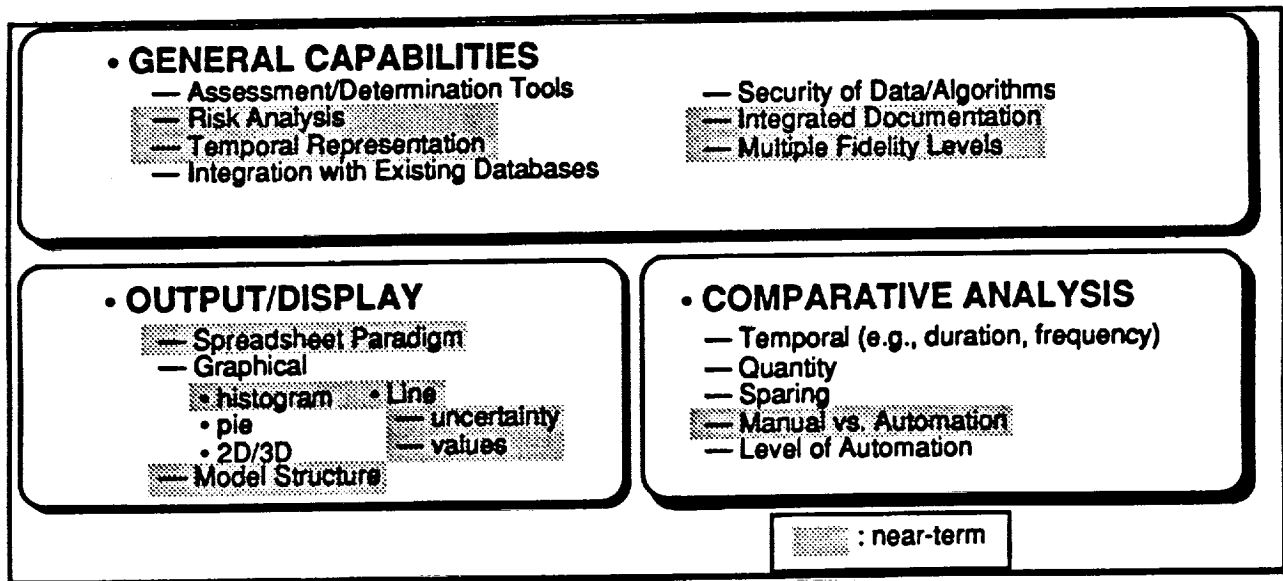


Figure 2. Near-term Objectives are an Achievable Subset of the Envisioned Capabilities of the Final ALCM Tool.

Within this contractual effort, the intention is to provide a proof-of-concept for the specified requirements of both the near- and far-term. As such, critical functionality will be demonstrated for this contract. Critical functionality will be agreed upon between NASA and Rockwell International and be within scope of this contract. The specific objectives for this contract are listed below:

- Provide conceptual development of ALCM with breakdown of modules
- Identify any required external modules
- Gather data appropriate to the capabilities of the model
- Construct algorithms to support model development, as necessary to demonstrate pertinent capabilities of the ALCM tool
- Begin development of graphical user interface (GUI)

4. Approach

4.1. Overview

The approach being used is consistent with both the scope and objectives of this contractual effort. The approach provides a demonstrable product that presents the envisioned capabilities of the ultimate ALCM tool. This enables feedback from the intended end users of the ALCM tool to gain their confidence and improve functionality and performance of the tool. Table 6 summarizes the steps to be taken.

- Assess desirable capabilities
 - structure into near- and far-term
- Identify useful existing models/data
- Identify parameters for utility analysis
 - cost
 - schedule
 - element
 - reliability
 - supportability
 - operability
 - interoperability
 - ...
- Define tool framework
- Encode scenario thread for model validation
- Provide transition path for tool development

Table 6. The Approach Ensures an Interim Product to Assist Tool Acceptance and Validity.

Underlying factors affecting the development of the ALCM tool include three analysis perspectives: cost, comparative, and decision. These facets are reflected in any efforts for this contract. By using these viewpoints, a more robust and useful tool is generated. Cost analysis is used in conjunction with task analysis to provide data generation and validation. Comparative analysis is used to analyze various possibilities for the use and timing of A&R in space systems. Decision analysis is used to explicitly treat uncertainty inherent in the problem domain. Table 7 provides a high level summary of the possible uses of these analyses.

- Cost Analysis
 - Parametric vs. Empirical
 - Confidence
 - Validation
- Comparative Analysis
 - Manual vs. Automation Alternatives
 - Automation alternatives
 - Phase/Component considerations
- Decision Analysis
 - Avoid "point" decisions
 - Identify key drivers
 - Relevant level of detail

Table 7. Utilizing Three Different Analysis Perspectives Provides the Opportunity for a More Robust and Useful ALCM Tool.

The identification, capture, and utilization of existing data and models are structured around the life-cycle of a space system. Correspondingly, an accepted and valid representation of this cycle is required early in this contractual effort for meaningful progress to occur. Figure 3 shows the definition of that life-cycle as delineated by the following three phases: research and development (R&D), investment and acquisition (I&A), and operations and support (O&S). This convention and nomenclature will be used throughout both this report and the entire contractual effort.

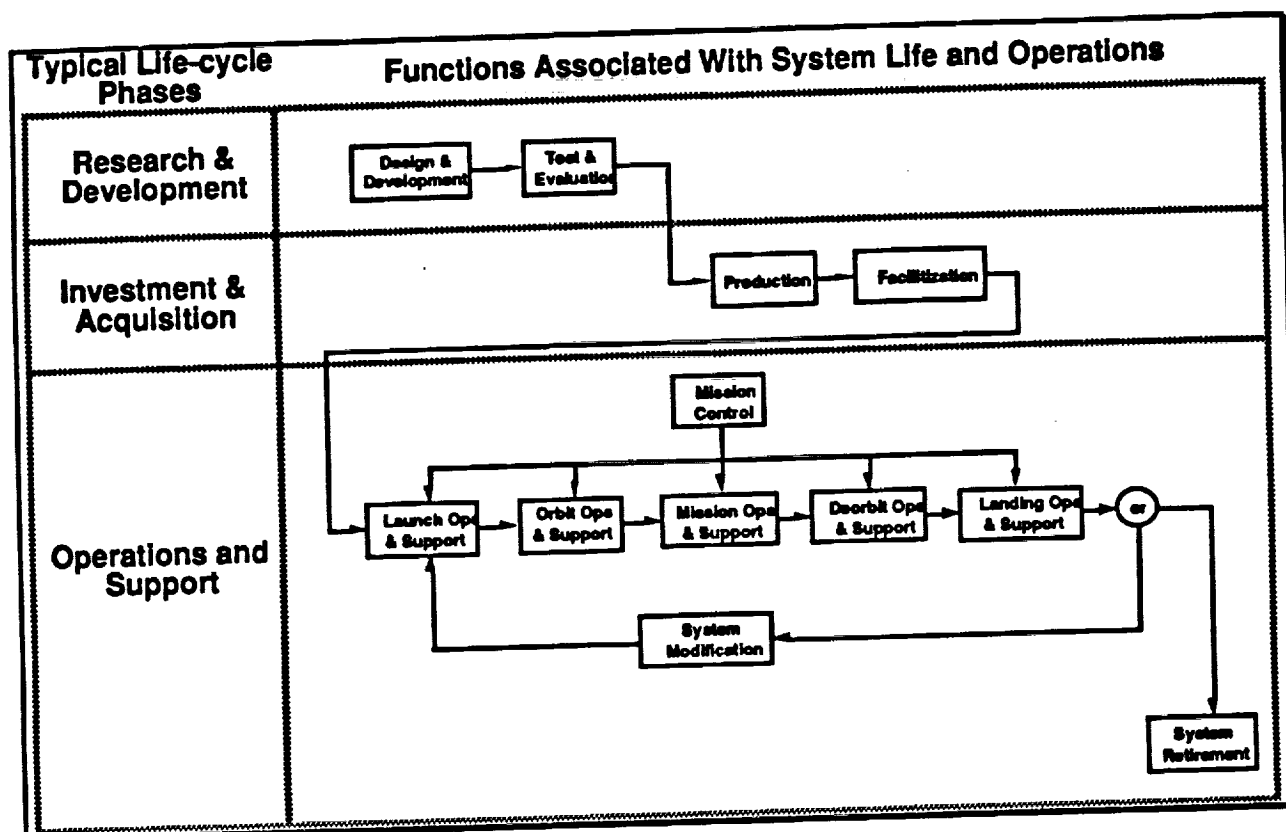


Figure 3. Program Life-cycle Phases and Functions for Space Systems Define the Modules for the ALCM Tool.

4.2. Existing Model Identification

Existing life-cycle cost models, relevant to A&R technologies, are surveyed to identify pertinent features that might be incorporated in the ALCM. These features are utilized to their utmost to minimize development risk and to maintain consistency with the general community. This survey also provides partial evaluation of each of the cost models/applications.

Existing models that might be assessed include, but are not limited to, the following:

- Brilliant Eyes Cost Model — United States Air Force
- Optimum Repair Level Analysis (ORLA) — NASA
- Production Rate Adjustment Factors (PRAF) — United States Army
- Atliss — United States Air Force
- LCC1 — North American Aviation (Rockwell International)
- Planetary Logistics Analysis and Evaluation Tool (PLANET) — Rockwell International

The existing models surveyed will be identified and comments noted in documentation. This report contains key features, attributes, and drawbacks (as applicable) for each of the models. Areas of applicability and usefulness for each of the models assessed are noted. Areas explored include:

- launch to orbit costs
- on-orbit replacement costs
- advanced technology usage projections
- man-tended systems costs
- unmanned systems costs
- advanced systems costs projections
- constellation systems costs
- user friendly man/machine interface (MMI) independent of user skills

A summary of findings from the above list of models is shown in Table 8.

LEGEND		Existing Models/Tools Potentially Applicable to ALCM			
M/R	Maintenance/Repair algorithms				
A	Approach				
SA	Spares Approach				
Preliminary Definition of ALCM Modules		Brilliant Eye Cost Model (U.S. Air Force)	Optimum Repair Level Analysis (ORLA - NASA)	LCC (Rockwell)	Planetary Logistics Analysis and Evaluation Tool (PLANET)
Design & Development		A			
Test & Evaluation		A			
Production		A			
Facilitization		A			
Launch Operations & Support		A	M/R,A		M/R
Orbit Operations & Support		A			
Mission Operations & Support			M/R,A	SA	M/R
Deorbit Operations & Support		A			
Landing Operations & Support			M/R,A		M/R
Disposal		A			

Table 8. Existing Models Are Leveraged to Accelerate the Pace of ALCM Tool Demonstration and Development.

While the models listed in Table 8 include a wide range of data and CERs (cost estimating relationships), direct reuse of those data and algorithms is highly limited. Ease of use and integration concerns have driven this analysis. The use of an existing approach as a basis facilitates model development. At the same time, the new model (i.e., ALCM) gains a legacy and increases user confidence since its basis is accepted information. The PRAF and Atliss models were not examined as planned. The utility gained from those models listed in Table 8 greatly decreased the additional returns gained from the PRAF and Atliss models.

There is no standard costing methodology for implementing a robotic system in commercial industry. Different robotic applications use distinct criteria for cost factors. For example, radiation in a nuclear reactor hot cell is the determining factor for cost justification, while in an automobile factory it may be the cost of production. Most performed cost analyses in these areas take into account the cost alternative options. Robotics is an option that normally distinguished from the others by its impact on the mission scenarios and unique benefits that it offers in the areas of task reassignment from to robot. Currently, there is enough data available to estimate DDT&E (design, development,

test, and evaluation) costs of robotic systems for commercial applications. However, the space program is lacking in this area in both ground processing and on-orbit applications.

Appendix 9.5 contains information on the generation of qualitative robotics CERs. Conversion of this expertise into equations for inclusion occurs in a follow-on ALCM effort.

Most robotics deployment and operation experience lies in the commercial sector. However, illustrative data points do exist for particular application as well as common off-the-shelf components. Appendix 9.6 contains these data. Initial complexity level definitions and estimates exist.

Likewise, NASA defines technology maturity with Technology Readiness Levels. Table 9 defines these levels.

NASA OAST Space Systems Technology Model (May 1980) Technology Readiness Levels	
Level 1	Basic principles observed and reported
Level 2	Conceptual design formulated
Level 3	Conceptual design tested analytically or experimentally
Level 4	Critical function/characteristic demonstration
Level 5	Component/breadboard tested in relevant environment
Level 6	Prototype/engineering model tested in relevant environment
Level 7	Engineering model tested in space

Table 9. Use of NASA Technology Readiness Levels Can Facilitate Risk and Development Cost Generation.

Recurring operations and support costs associated with a given system typically dominate life-cycle costs. In fact, operations and support comprise 60% of the total life-cycle cost.[ilsg] Main constituents of these costs are maintaining spares' inventories and supporting a maintenance and repair infrastructure. The greater diversity of components, and more frequently items fail, then the greater will be the recurring operations and support costs. Therefore, it is vital to LCC to be able to estimate the failure rates of the components comprising a system.

The Planetary Logistics Analysis and Evaluation Tool (PLANET) is a model, developed for use on NASA space systems, to predict failure rates for space systems equipment. It requires only minimal equipment definition data, that which is typically associated with conceptual phases of a program.

It is a data base management system model, with several relational files that perform:

- determination of failure rates
- estimation of maintenance demands over time
- aggregation of maintenance demands for various systems/subsystems
- sensitivity and tradeoff analyses
- output of results in tabular or graphical formats

Many of the features go beyond the ALCM requirements, in that the ALCM already has imbedded many of the features already or does not need them in the LCC analysis. It would appear that the centerpiece of the PLANET model, those files that determine failure rates, would be its sole contribution to the ALCM. Attaching simple spreadsheet files to the ALCM may accomplish this extension. Read and write protection of these spreadsheet files is a vital requirement since the contained data would be Rockwell proprietary algorithms.

Appendix 9.5 contains information on the generation of qualitative robotics CERs. This expertise is converted into equations for inclusion in the follow-on ALCM effort.

4.3. Existing Data Source Identification

The task consists of the effort to identify data available within Rockwell International applicable to automating space systems. Currently known costing data that might be researched and assessed include, but are not limited to, the following:

- Rockwell Shuttle Operations Company (RSOC) task manpower database
- Rockwell International Space Systems Division Shuttle DDT&E cost database
- Personnel Launch System (PLS) complete life-cycle cost analysis done by Rockwell International Space Systems Division (SSD) for NASA Langley
- Rockwell (Independent Research and Development) IR&D project for post-landing thermal tile inspection process for the Space Shuttle

A variety of data sources were examined during this period to identify information (either for use as CERs, or as constant-type variables) that would be appropriate for inclusion in the ALCM. The data sought, fell into two categories: (1) methodologies and/or algorithms related to parametric cost estimating/predicting, and (2) historical and/or accounting data that would enable the development of CERs.

Seven different data sources were researched, besides the models and data bases that were evaluated (refer to paragraph 4.3). A summary of findings is contained in Table 10.

Source Name	Authority	General Content	CERs	Accounting Historical Data	Applicable To ALCM?
Earth To Orbit Support System Concept Study	SSD Internal Research & Development (Project 90237)	Contains ground processing and ground operations CERs related to different launch vehicles	✓		YES
Humans/Automation/Robotics/Telerobotics Study	NASA/JSC Report #: JSC 45105	Contains operating time, manhours for monitoring and maintaining robotic systems for a Lunar Base relative to SEI Option 5A		✓	YES
MERGER Data Base	Rockwell Shuttle Operations Contract (NASA/JSC)	Contains actual support manpower expenditures to support shuttle flight design, mission planning, etc.		✓	POSSIBLY
On Orbit Assembly/Servicing Task Definition Study and Advanced Automation For In-Space Vehicle Processing Study (Combined Studies)	McDonnell Douglas Study For NASA/KSC (Contract #: NAS10-11400)	Contains task times and trades, between human and robotic assembly and processing of vehicles in space		✓	YES
PLS LCC Estimates	NASA/LaRC Study (Contract #: NAS1-18975)	Contains Both Accounting and CER based estimates, resident in RCA-Price model, for light weight reusable manned vehicle	✓	✓	YES
Shuttle DDT&E Data Base	SSD Internal Data Base	Contains CERs for large mass launch vehicles and launch vehicle subsystems	✓		NO
Tile Rewaterproofing Study	SSD Internal Research & Development (Project 90237)	Accounting type data relating costs associated with the various elements of the Tile Rewaterproofing Robotic System		✓	YES

Table 10. Summary Findings of Data Sources Reviewed.

4.4. Advanced Technology Impact Determination

This task includes the efforts to identify existing means for extrapolating advanced technology transfer to space systems. This may include the following efforts:

- identification of existing methodologies within the industry
- identification of existing methodologies within Rockwell International's Space Systems Division
- modification of potentially applicable methodologies from other life-cycle costing disciplines

When predicting costs for advanced technology systems, it is necessary for a model to be capable of trading varying levels of technology readiness, not only as an element of cost, but also as a risk or uncertainty factor. When addressing robotic technologies for space applications, there are three factors that drive technology requirements.

- (1) External/environment factors (i.e. atmospheric pressure, radiation exposure, humidity, ambient light levels, etc.) impose new technology requirements on emerging robotic systems.
- (2) As robotics (in general) mature, requirements are being imposed on them to possess greater versatility and utility, and to perform tasks faster, better, with greater precision. This implies greater robotic autonomy, as well as improved performance in myriad areas (i.e., greater payload capability; self-healing; faster response time in both manipulators and end effectors; improved visual acuity; and increased tactile sensitivity.)
- (3) Related to, but beyond Factor #2, future robotic systems will need to be more adaptive and more interactive with their dynamic operating environments. To accomplish this, robotic systems will take advantage of the emerging neural network technology, and continued progress in knowledge based systems, to enable more human-like heuristic learning and deductive reasoning capabilities.

4.4.1 Assessment of Technology Drivers and Technology Readiness Levels for Robotic Systems.

First, external/environmental factors are driving robotics technology requirements by exposure to phenomena that, in some cases, go well beyond the benign and ambient environment of earth's surface. In most cases these environmental factors drive material selection (i.e., radiation hardening for nuclear applications; corrosion resistance and high strength requirements for underwater applications.) They can also drive other robotics subsystem technology requirements, such as better visual acuity for low ambient light conditions, as in space and underwater applications.

In response to the second factor, technology improvements imposed by new performance requirements vary widely depending on the robotic systems' application. It is best to assess these technology requirements by robotic subsystem.

4.4.1.1 Mobile Base Platform

Platforms with different categories of payload capabilities and mission requirements will vary in their power train, chassis, suspension and braking subsystems; as well as their positioning accuracy subsystems, and control and navigation subsystems.

-
- (a) **Power Train, Chassis, Suspension and Braking Subsystems** – This particular subsystem is characterized by relatively mature technologies—due to robotic system, military, construction industry, and automotive industry developments and requirements. This is not meant to say that levels of complexity may change. Volume constrained spacecraft will impose special packaging requirements and terrain considerations may drive totally independent drive, suspension, and braking systems.
- (b) **Positioning Accuracy Subsystems For The Mobile Base Platform** – Again, this subsystem is characterized by relatively mature technologies—due to developments in the air/ground segments of the military and commercial aviation industries, as well as work done in robotics fields. As accuracy requirements increase, different and/or integrated navigation systems will be employed to meet the particular requirement, which is more of a complexity issue.
- (c) **Control and Navigation Subsystems** – Control and navigation of the mobile base/platform can be achieved by one of three methods: (1) teleoperated/telepresence, (2) self guided, and (3) intelligent mobile base/platform. Control and navigation technologies are still maturing, and in some cases are only in the breadboard stage. Some of the driving technology factors are:
- data transmission rates
 - integration of robotic sensory data, health and status data, as well as performance and relative position data
 - distances between remotely operated control stations and the robotic system
 - detection and avoidance of known and unknown objects
 - autonomous path planning, utilizing preprogrammed topographic maps and 3-D imaging systems
 - operating and maneuvering in a dynamic environment (where other objects in close proximity to the robotic system are also moving)

4.4.1.2 Manipulator Subsystems

There are five separate components characterized by their own respective technology development schedules, they are: (1) Degrees of Freedom (DOF), (2) Reach, (3) Payload Lift Capability, (4) Manipulator Positioning Accuracy, and (5) Manipulator Command and Control. A combined discussion follows due to the synergistic relationship between DOF, Reach and Payload Lift Capability. Similarly, Manipulator Positioning Accuracy, and Command and Control will be addressed in aggregate.

- (a) **DOF, Reach and Payload Lift Capability**—Technologies in these areas are, again, relatively mature due to robotics developments in other areas. In the vacuum and zero-gravity environment of space however, technological advancements are still required for bearings and lubricants at the manipulator joints. Design complexity is usually the driving factor for these robotic system capabilities. To achieve greater payload lift capability, generally increased motor power is required and larger manipulator arm linkages or higher strength-weight materials are required. Similarly increasing reach distances requires increased mechanism weight and control system complexity.
- (b) **Manipulator Positioning Accuracy and Repeatability, and Command & Control Subsystems**—Higher accuracy is a function of mechanical design, sensor systems and control. There are a wide variety of control architectures and algorithms that are used for control of manipulator systems. They vary from simple to adaptive model reference control systems capable of optimizing system performance. Additionally, interfaces between operator and the machine could also vary drastically depending on the requirements.
1. **Preprogrammed**—This is the most primitive level of interaction between man and robotic manipulator arms. The robot does not interact with the operator and its interaction with the surroundings is extremely limited. This is a relatively mature approach, depending on the task performance.
 2. **Teleoperated/Telepresence**—Here, the manipulator system is controlled by a master arm or a joystick. Usually there is a one-to-one correlation between the master arm and slave arm degrees of freedom. Some of the more complicated systems may also detect the force at the slave and feed a portion of that back to the operator and master arm. This and the following three methods have varying levels of technology readiness depending on their application (or intended application, i.e., systems that are not yet operating in their intended environment, such as space robotic systems), and the complexity of the tasks they are required to perform. Some of the areas where technology development is required are listed in the Mobile Base Platform section.
 3. **Supervisory Control**—The manipulator arm is capable of decomposing low level tasks into subtasks. The operator interfaces with the robot through higher level commands and can supervise the operation with manual control and emergency shut down capabilities.
 4. **Coordinated Motion**—This system involves multiple manipulators that have to be coordinated to avoid damage to the work piece and/or the arms. Depending on the task and controller requirements, forces and torques may have to be minimized in one or more axis, which requires the a complex network of sensors to detect the forces and torques in each axis.
 5. **Task Based Control**—This is the highest level of operator interface. Task objectives are given to a robot work cell controller. The workcell controller decomposes these objectives into tasks, subtasks and motions and executes them accordingly. If an anomaly occurs, the controller can develop a contingent set of subtasks and execute them to work around the problem(s).

4.4.1.3 Special Tools/End-effectors

End effectors and end effector technology readiness are function/task dependent. Therefore, any attempt to create a taxonomy of end effector technology readiness would be purely subjective and obsolete in a short time. However, their respective purposes/functions could be generically categorized to assist in understanding what is driving system costs. These categories are type of end effector (contact vs. non-contact), force/torque & micro positioning control, and machine vision.

(a) Type of End Effector

- **Contact End Effector**—Here the end effector is used to grasp or perform a function on an object. The most basic end effector is the parallel jaw, which is widely used for a variety of applications. Power tools such as screw drivers are another type of end effector. As the tools become more specialized, technology readiness can decrease.
- **Non-contact End Effector**—In this instance, the end-effector moves an attached sensor system package to perform non-destructive testing (NDT) or monitor an on-going operation. Typically they carry a fixed mass. More complicated systems may be required to guide probes through maze-like systems, which may require more system flexibility and degrees of freedom.

- (b) **Force/Torque and Micro Positioning Control**—Most end effector control systems are included as part of the manipulator arm overall control scheme. However, those which offer multi-DOF often possess their own closed-loop control system. These end effectors are used either for operations where delicate objects are handled or they are used as micro positioning systems (the re-waterproofing end effector serves both these purposes). In each case the end effector design becomes more complicated and more sensors have to be used to control forces, torques and positioning accuracy. Depending on the application, significant technological development is required with an accompanying dramatic increase in cost.

- (c) **Machine Vision**—Vision systems can be independent of the end effector. However, in some applications they are part of the end effector. Most vision systems currently used are off-the-shelf and operate under specific operating environments. These vision systems can be integrated into a system easily and their costs are relatively modest. Costs for vision systems increase as they migrate to unstructured environments. Sample vision system applications include:

1. **Inspection and monitoring**—Typically used to locate and identify specific features under a varying light/shade and orientations. This type of vision system capability may not be available off-the-shelf and could require a sizable development cost.
2. **Guidance, path planning and collision avoidance**—Along with the aforementioned, interpretation of images and map generation (2-D and/or 3-D) capabilities could be required in this type of vision system. This type of requirement is characterized by a low level of technology readiness, and development of this capability will require a substantial research and development investment.

Finally, technology related to imbuing robotic systems with more human-like cognitive capabilities is presently 'over the horizon' and is very dependent on what happens in the neural network and advanced logic technology arena.

4.4.1.4 Rockwell Space Systems Division Involvement in Robotics and Related Technologies

The robotics group at SSD has a laboratory with a number of different systems and end effectors for performing tests and a wide range of analyses. They have supported development of robotic systems requirements for space applications and have been active participants in NASA robotics related activities. They also have worked with the Oakridge National Laboratory, in developing robotics requirements and systems related to the nuclear power industry and other public and private institutions. Members of this group include:

- co-chairman of the American Institute of Aeronautics and Astronautics (AIAA) Robotics Standards committee
- participants on the Humans/Automation/Robotics/Telerobotics (HART) Working Group
- participants on the NASA/DoD/Industry Space Assembly and Servicing Working Group

4.4.1.5 Identification of Existing LCC Methodologies

The SSD robotics group has been involved in the development of a number of robotic systems, including the Tile Rewaterproofing System for NASA-KSC. In each case they have access to actual cost data, however, they have not had the requirement imposed on them to aggregate the data in such a manner to enable them to turn the accounting data into CERs. The life-cycle cost of a robotic system is compared with that of other alternatives. However, it is as important to quantify the benefits that different options offer to the operation or mission scenarios and include them in the LCC model.

$$\text{Net Cost} = \text{SUM}(\text{DDTE \& Operation Cost}) - \text{SUM}(\text{Quantified Benefits})$$

This will require development of an end-to-end cost dynamic model of the overall Program (system and/or environment). This model should allow for an overall Program cost calculation over a given period. It should also be able to calculate the cost and benefits associated with different Program components. Such a cost model could be used to evaluate cost impact of a system change over the whole Program. During Phase B of the Space Station Freedom, NASA JPL proposed the development of a similar cost model and published a number of papers on the subject. Furthermore, Rockwell during the same period, concentrated on the development of techniques to isolate and evaluate system net costs. Here, the objective was to identify a set of boundary conditions for a given system that allows the inclusion of outside factors to the system cost. Both techniques along with others offer the building blocks for a comprehensive LCC Model. As more cost data becomes available, different portions of the model could be tested and verified.

4.5. Decision Modelling System (DEMOS) Tool Capabilities

Decision Modelling System, DEMOS, is an interactive environment for structuring, analyzing, and communicating probabilistic models.[4] It allows the construction of problem domain mathematical representations with explicit treatment of uncertainty. Variables defined within DEMOS can be either deterministic or probabilistic. The DEMOS computational engine is capable of handling a wide variety of functions in both regimes of variable types, including their various combinations.

DEMOS executes on Macintosh computer hardware platforms and, therefore, conforms to easily usable user interface conventions and protocols. DEMOS was constructed to be highly interactive and tailorable. Table 11 summarizes its key features.

- Influence diagrams display model structure
- Hierarchical structure helps organize large models
- Any uncertain value can be probabilistic
- Graphic parametric and probabilistic uncertainty analysis help identify key sources of uncertainty
- Non-procedural modelling language reduces programming effort
- Array abstraction allows flexible construction of multidimensional models
- Hypertext model with integrated documentation supports collaboration and sharing of models
- Libraries of functions support customizing for particular classes of application

Table 11. DEMOS is a Highly Flexible and Appropriate Environment for the Solution of LCC Problems.

Given these general capabilities, a mapping has been constructed of pertinent DEMOS features and their envisioned utilization for the ALCM tool. Plans include incorporation and integration of existing data and models into the tool. The demonstration developed during this contractual effort attempts to provide a typical usage of these pertinent features. In this way, the power and flexibility of DEMOS can be used to demonstrate the validity of the ALCM tool. Table 12 summarizes the intended usage of DEMOS features.

DEMOS FEATURE	ENVISIONED USAGE
HIERARCHICAL STRUCTURE	Defined model structure
PROBABILISTIC VARIABLES	Used for highly uncertain inputs
UNCERTAINTY ANALYSIS	Identify cost drivers
MODELLING LANGUAGE	• Provide function flexibility • Complex mappings of elements in the life-cycle
ARRAY ABSTRACTION	• Multidimensional trade space • Incorporation of existing data
INTEGRATED DOCUMENTATION	Structured presentation of critical information (validation, heritage, explanation, etc.)
LIBRARIES	Development of a tailored library of functions explicitly constructed for the ALCM tool

Table 12. DEMOS Features Can Be Easily Tailored for the ALCM Tool.

5. Automation Life-cycle Cost Model (ALCM)

5.1. Overview

The ALCM is a life-cycle cost analysis framework developed using the DEMOS application. It provides parametric and accounting capability for development, acquisition, and operation and support phases of the life-cycle. Figure 4 shows the ALCM methodology flow of the life-cycle cost process.

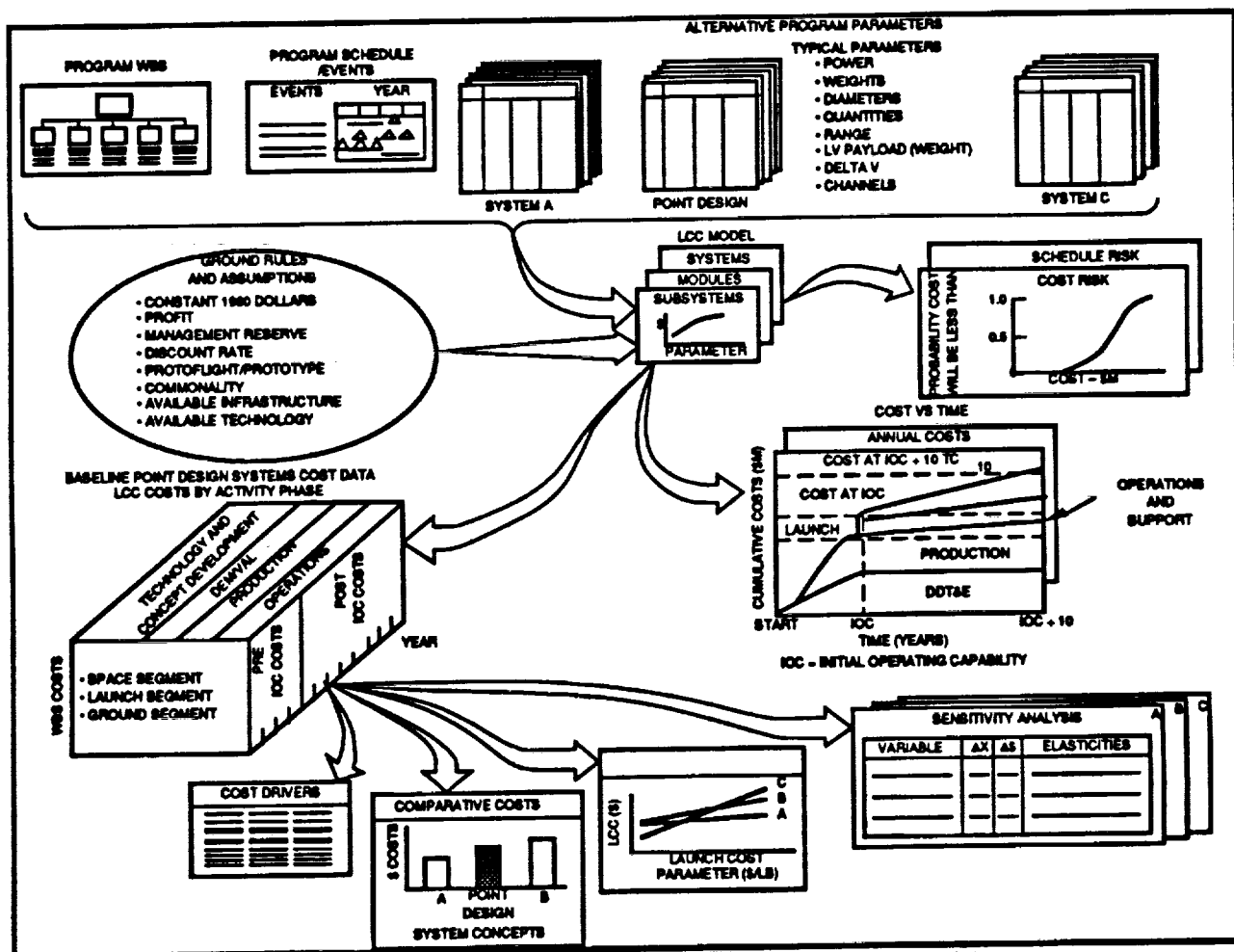


Figure 4. Life-cycle Cost Methodology Flow Interfaces With All Aspects of a Program.

Development of an LCC estimating/prediction model requires a rigorous effort to capture all data relevant to the program and system being modeled. If used to influence system design and operations decisions, the model must have the inherent capability to allow designers/analysts to perform, in real-time, subsystem/element cost tradeoff/sensitivity analyses. Finally, the LCC model needs to have the flexibility to "report" costs from various perspectives, which reflect different customer perspectives, i.e., program managers, ground segment managers, spacecraft segment managers, risk management personnel. Figure 4

illustrates the program and system inputs, engineering economic analysis capability and reporting capabilities contained within the ALCM.

The model needs to be able to capture the breadth of program definition and various engineering opportunities. In a generic model, life-cycle phases are easily modeled, in the sense that the model only requires quantification of respective phase durations. Similarly, the hardware/software system is relatively easily modeled—either specific CERs or accounting based procedures, within the tool, can be employed to estimate unit costs. Other program specific ground rules and assumptions are modeled by providing place-holders for such things as discount rates, labor rates, etc. Another required input, but difficult to model generically, is the program work breakdown structure (WBS). Every program has its own unique hierarchical WBS. However, at the higher levels they all generally contain the same elements, though sometimes by different names. The generic cost categories modeled in the ALCM, which is consistent with the higher levels employed on previous NASA and DoD programs, are defined in Section 5.2.1.

To perform engineering economics analysis requires the model be constructed to handle hardware/software system metrics that define and "shape" non-recurring and recurring costs. The ALCM is generic in the sense that it provides a capability for the user to enter CERs and engineering/integration relationships. Accordingly, a designer will be able to enter such data as, unit weight, power requirements, number of pixels, etc., and determine a specific item cost and assess the change to the baseline system LCC. It is important, if the ALCM is going to assist in making design and operations decisions, to perform a number of different types of analyses. The DEMOS environment facilitates performance of the following engineering economics and LCC analyses:

- Identification of cost drivers
- Depiction of risks (e.g., uncertainty in numeric values)
- Performance of comparative analysis
- Performance of cost/benefit analysis

Following program and system definition, the ALCM is capable of quantifying and reporting costs in a number of different formats and perspectives, depending on the particular user and need. The aggregate costs can be viewed from the following perspectives:

- Cost-risk
- Costs by program phase
- Costs by WBS
- Hardware/software costs (e.g., to support Design to Life-cycle Cost [DTLCC])
 - by program phase, and/or
 - by WBS
- By any combination of the above

5.2. System Architecture Definition

The ALCM architecture provides for generic life-cycle cost categories and specific program work breakdown structures (WBSs).

5.2.1. Generic Life-cycle Cost Categories

Figure 5 depicts the phases of a life-cycle in the ALCM. The Operation and Support phase includes the disposal of the unit.

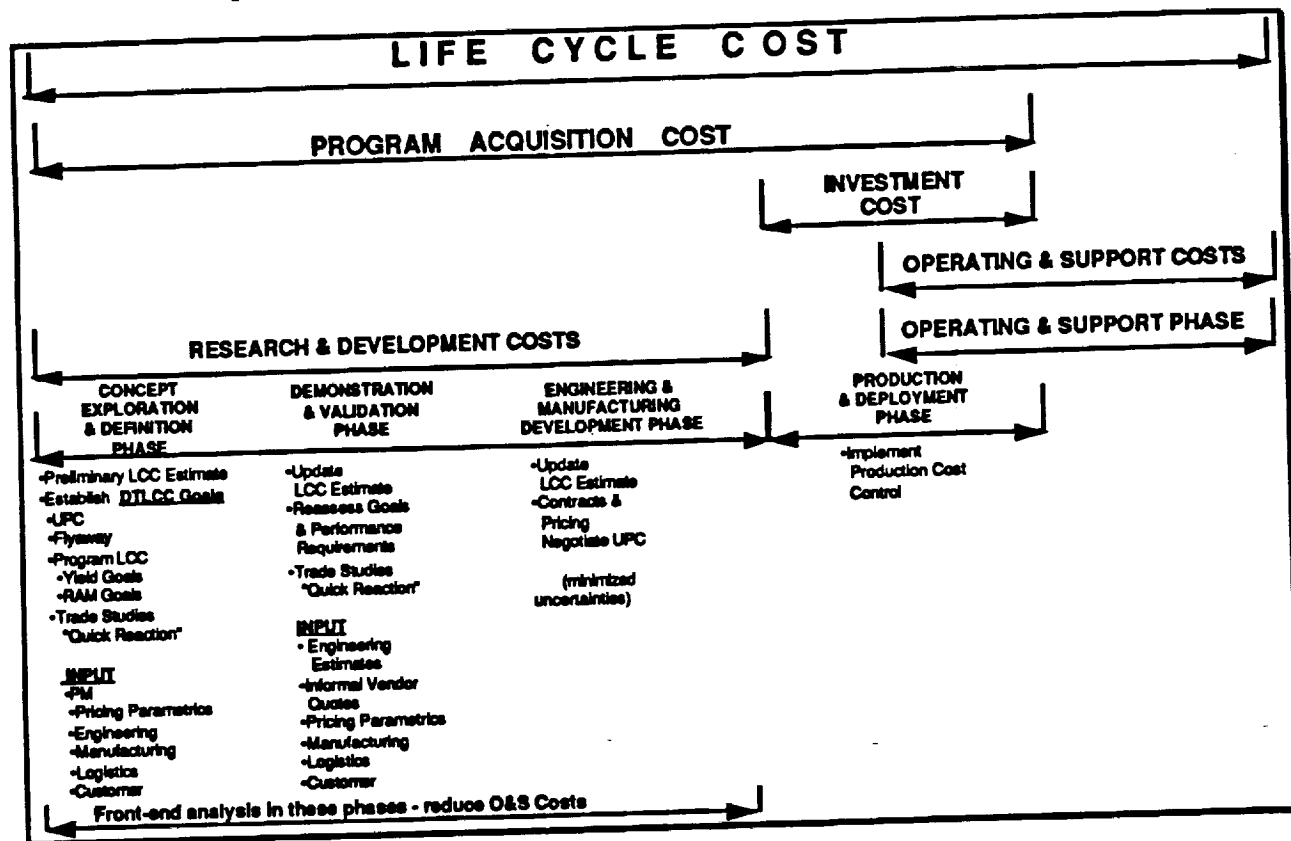


Figure 5. ALCM Life-cycle Phase Definitions.

5.2.1.1. Research and Development (R&D) Phase Cost Categories.

Table 13 lists the R&D cost elements. This phase of the life-cycle includes Definition (Phase 0), Demonstration & Validation and the Engineering and Manufacturing Development phases of R&D. The categories shown in Table 11 may be expanded to a lower level if the program requires.

PRIME APPROPRIATIONS	DEFINITION REFERENCE	COST ELEMENT
	1.0	RESEARCH AND DEVELOPMENT
RDTE	1.01	DEVELOPMENT ENGINEERING
RDTE	1.02	PRODUCIBILITY ENGINEERING AND PLANNING (PEP)
RDTE	1.03	TOOLING
RDTE	1.04	PROTOTYPE MANUFACTURING
RDTE	1.05	DATA
RDTE	1.06	SYSTEM TEST AND EVALUATION
RD/OM	1.07	SYSTEM/PROJECT MANAGEMENT
RD/OM	1.08	TRAINING
RD/MC	1.09	FACILITIES
RDTE	1.10	SOFTWARE
RDTE	1.11	OTHER

Table 13. Research and Development Phase Cost Category Definitions.

5.2.1.2. Investment and Acquisition (I&A) Phase Cost Categories.

Table 14 identifies the cost categories included during this phase of the life-cycle. These costs include production non-recurring and recurring costs, initial deployment and other initial investment costs. The categories shown in Table 12 may be expanded to a lower level if the program requires.

PRIME APPROPRIATIONS	DEFINITION REFERENCE	COST ELEMENT
	2.0	INVESTMENT AND ACQUISITION
PR/MC	2.01	NON-RECURRING INVESTMENT
PROC	2.02	PRODUCTION
PROC	2.03	ENGINEERING CHANGES
PR/OM	2.04	SYSTEM TEST AND EVALUATION
PR/OM	2.05	DATA
PR/OM	2.06	SYSTEM/PROJECT MANAGEMENT
PR/MC	2.07	OPERATIONAL/SITE ACTIVATION
PR/OM	2.08	TRAINING
PR/OM	2.09	INITIAL SPARES AND REPAIR PARTS
PR/OM	2.10	TRANSPORTATION
PR/OM	2.11	FACILITIES
PR/OM	2.12	OTHER

Table 14 Investment and Acquisition Phase Cost Category Definitions.

5.2.1.3. Operation and Support (O&S) Phase Cost Categories.

These costs begin at initial operating capability (IOC) and continue through disposal of the unit. The cost categories apply to launch operations and support, orbit operations, mission operations and support, de-orbit operations, and landing operations and support and disposal. The categories shown in Table 15 may be expanded to a lower level if the program requires.

PRIME APPROPRIATIONS	DEFINITION REFERENCE	COST ELEMENT
	3.0	OPERATION AND SUPPORT
	3.01	PERSONNEL
	3.01.1	CREW PAY AND ALLOWANCE
	3.01.2	MAINTENANCE PAY AND ALLOWANCE
	3.01.3	INDIRECT PAY AND ALLOWANCE
	3.01.4	FLIGHT PAY AND MISC
	3.02	MISSION OPERATIONS AND SUPPORT
	3.03	ORBIT OPERATIONS AND SUPPORT
	3.04	DEORBIT OPERATIONS
	3.05	LANDING OPERATIONS AND SUPPORT
	3.06	MAINTENANCE
	3.06.1	LABOR
	3.06.2	MATERIAL
	3.06.3	TRANSPORTATION
	3.07	SPARES
	3.08	TRAINING
	3.09	DOCUMENTATION
	3.10	SUPPORT AND TEST EQUIPMENT
	3.11	SOFTWARE MAINTENANCE
	3.12	DISPOSAL
	3.13	FACILITIES
	3.14	OTHER

Table 15. Operation and Support Cost Category Phase Definitions.

5.2.2. Specific Cost Categories

Within each of the generic modules of the model, by phase, the specific program WBS that applies to that phase of the program will be tailored to fit the program being analyzed. Figure 6 shows an example of the format that will be seen on an output report from the ALCM.

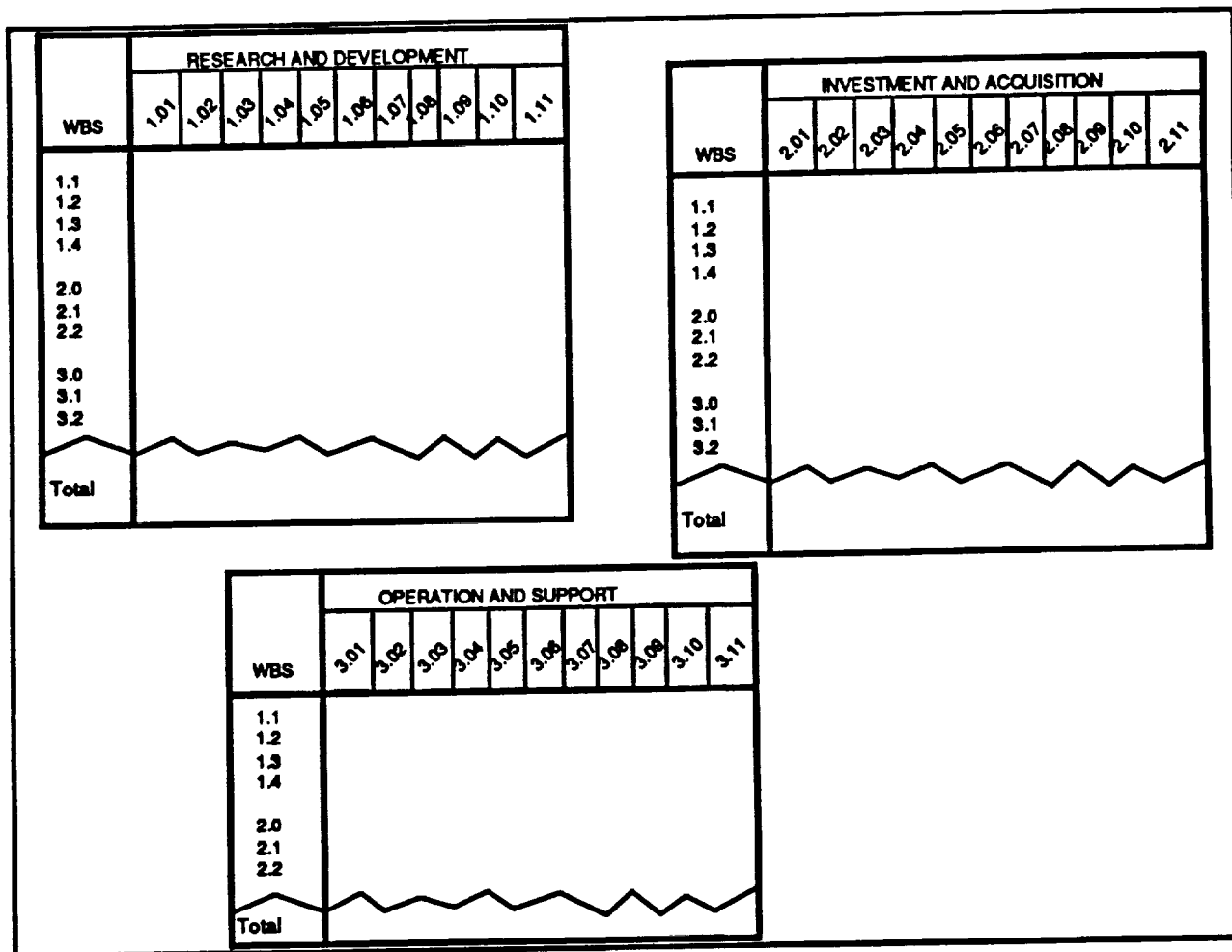


Figure 6. Sample ALCM Work Breakdown by Life-cycle Cost Phase.

5.3. Model Development

Model development was driven by the long-term objectives of this effort within the system architecture definition (as specified in Section 5.2). Traditional life-cycle costing techniques were used to specify the model content. The ability to use both top-down and bottom-up approaches within the model was an objective. The ability of the model to respond flexibly to differing requirements and situations of various programs was a stressing requirement.

The decision was made to utilize a well-known and understood application to drive the development of the model. Rockwell's Space Systems Division's experience with robotic applications regarding the Space Shuttle ground processing met these requirements. Not only are the current manual processes well within the scope of activities for the Corporation, the proposed automated improvements include vital Rockwell components. Access to key data was eased. The principle investigator for the robot end-effector on these applications participated in this contractual effort. Her insight, not only of Rockwell participation, but of the effort as a whole greatly contributed to the model development.

Subsequent subsections provide background information and the decision making process that were used to consider and, finally, select the tile waterproofing application as the driving situation for our model. Its apparent extensibility to space-based systems was taken into account using current contractual knowledge of flight robotic systems and proposed efforts for future space mission (e.g., Moon and Mars expeditions).

5.3.1 Background

Boeing Company conducted a study to identify and evaluate potential automation applications to improve Space Shuttle ground processing between flights at the Orbiter Processing Facility (OPF). One hundred and thirty-eight tasks were identified for improvement in the current process. The top eight were prioritized and described in detail. Three of these are as follows:

1. **Tile Re-waterproofing** — Over 22,000 ceramic tiles are used in the Space Shuttle Thermal Protection System. These tiles are precisely installed and their operability is altered by environmental conditions prior, during, and after a Shuttle flight. An effective method of tile protection is waterproofing. The chemical used for this protection is consumed during the flight, so it must be replaced.

The current method involves dimethylethyloxysilane, which is a toxic compound, and is highly labor-intensive. The current process now takes about five days and must be executed during the third shift for safety considerations.

A robotic system has been proposed and currently is in various stages of development. The system involves a mobile base, manipulator, vision system, tile re-waterproofing end-effector, and workcell controller.

Design and proposed development costs are known for system for this application. Projected O&S costs are also known.

2. **Tile Step and Gap Evaluation** — Flight regime environmental impacts on the Shuttle thermal tiles can cause movement from their intended positions. Step is the surface-to-surface height between adjacent tiles. Gap is the width of the space between adjacent tiles. Evaluation of Step and Gap information of over 22,000 tiles is required after each flight. More attention is concentrated on the height of the leading edge wing tiles, however.

The current method is a highly labor-intensive process. A proposed solution involves most systems for the tile re-waterproofing application with the exception of a different end-effector. This solution has been conceptualized and is not as mature as the re-waterproofing application.

3. **Tile Bonding Evaluation** — The current evaluation method to determine the integrity of the tile bond to the Shuttle is a highly labor-intensive process and uses potentially destructive actions. A proposed solution includes the use of laser shearography implemented on an end-effector on the robotic system noted in the re-waterproofing application. This solution has been conceptualized, and demonstrated in the Rockwell Palmdale facility on Orbiter 102, but is not as mature as the re-waterproofing application progress.

5.3.2 Approach

The family of terrestrial applications listed in the previous section provides many advantageous elements to our approach for model development. Use of a ground process for a space system offers concrete cost values and extensive experience. This information is preferred to permit the development of confidence in an LCC model. The use of accepted data and methods, coupled with recognized experts, provides a means for extrapolating implementation in non-terrestrial deployed systems.

Figure 7 graphically presents the general approach taken for model development and is summarized in the following list:

- Waterproofing application is used to generate CERs
- Resultant model is used to predict Step and Gap and/or bonding application costs
- Use of PLANET data and modelling capabilities will be implemented within the ALCM framework to provide vision to far-term space system modelling.

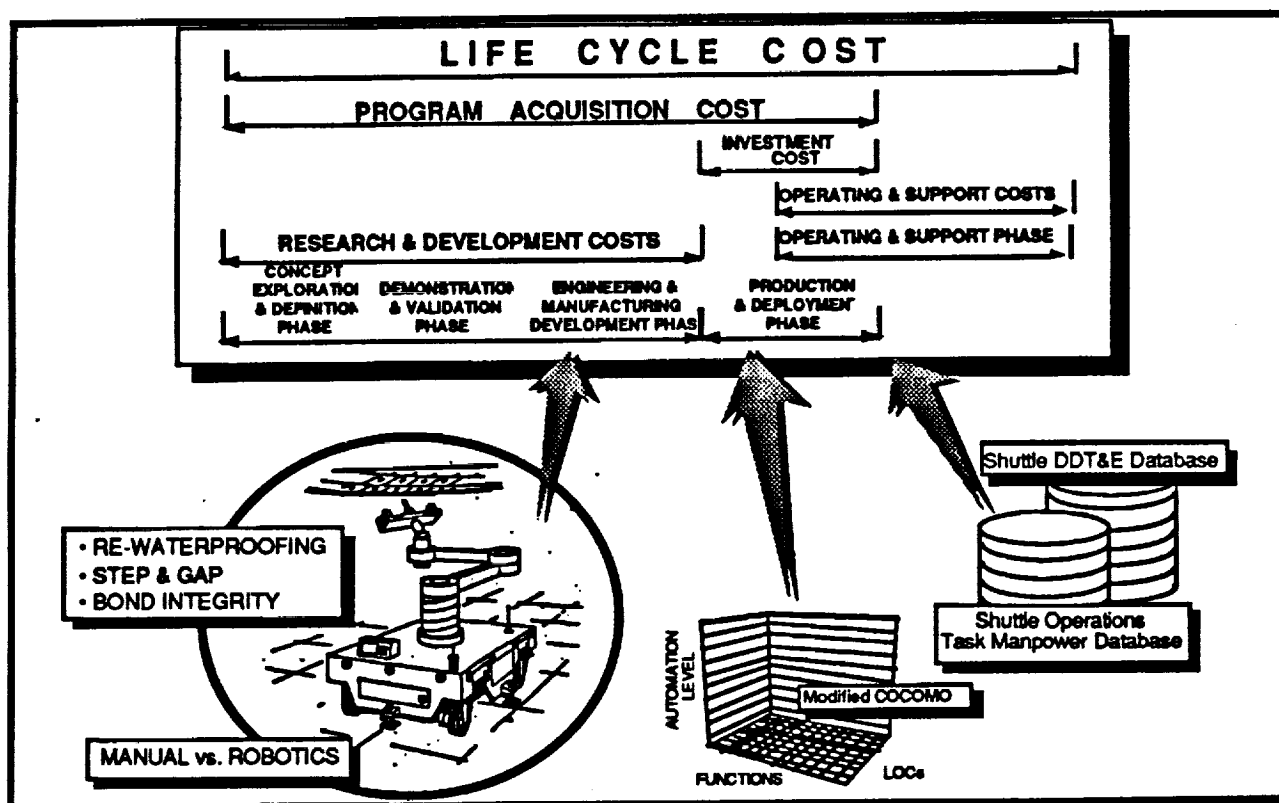


Figure 7. The ALCM Approach Capitalizes Upon a Large, Known Body of Pertinent Data, Advanced Robotic Applications, and Standardized Costing Methods to Address the Entire Life-cycle Cost for Space Systems.

This approach permits the generation of CERs that can be readily validated using internal Rockwell knowledge and experience coupled with data available from the NASA center sponsoring the robotic effort (NASA Goddard). Given the close relationship of the tile re-waterproofing, Step and Gap, and bonding applications, it was determined that the CERs developed from the first application would be appropriate for the other two. Since those efforts are currently in progress, or very nearly so, the urgency and realism provided increases confidence in the ALCM results. Also, known space-based robotic efforts can be related to this suite of applications to generate a legacy for ALCM costing of flight systems.

The PLANET model, generated under NASA contract NAS9-18344 (Planetary Surface Systems Maintenance Assessment Study), addresses the logistical aspects of deploying manned bases in non-terrestrial environments. This model generates O&S reliability and maintainability data that will feed into the ALCM model. These data are used to derive cost figures for that phase. The integration of the PLANET model into the ALCM framework is outside the scope of the current effort. However, preliminary investigations have been conducted into the viability of that integration. There appears to be no great technical risk associated with that extension of the ALCM framework.

An explicit requirement made early in this contract was that the construction of the ALCM framework be based on generally accepted practices and principles within the LCC domain. The following list contains the explanation of this requirement:

- This methodology includes the following items and relationships:
 - **Life-cycle Phase Definitions** — This definition includes all costs associated with a program, from conceptualization through disposal. Program development (including research and development), acquisition (investment), and operating and support costs comprise life-cycle cost.
 - **Program WBS**—Identifies and defines elements of a system for all pertinent phases of the overall life-cycle. This structure is typically hardware-oriented. The structure is a hierarchical tree.
 - **Cost Categories and Elements**—This representation is accounting-based. Accepted elements are defined for the Research and Development, Investment and Acquisition, and Operation and Support phases.
 - **Mapping of WBS to LCC Phase**—The matrix defined by relating WBS elements to cost elements is always program-unique. However, generic WBSs and cost elements are known.
- Automation techniques are to be applied during the system design definition. Automation of systems, subsystems, components, modules, and line replaceable units (LRUs) is possible. However, experience indicates the subsystem level is sufficient for most situations. The representation of automation capabilities is a continuous function. However, its definition within this version of the ALCM framework will be a discrete function.
- The ability to specify and revise CERs is crucial to the intended operability of the ALCM framework (and eventual tool). A minimal set of 3 formats is supported.
 - 1) **Parametric CER**—This CER is experience-based and expressed in alternative program parameters. This relationship has been derived through iterative analyses of previous systems guided by "technical analogy." That is, similar elements within previous systems are used as the independent variable while other terms are varied and graphed. A suite of these graphs is analyzed to produce the CERs.
 - 2) **Accounting-based**—This format is similar in intent as the parametric CER. However, the terms in the equations are extracted from accounting data and production rates.
 - 3) **Throughput (vendor quote)**—This format typically has the simplest form of the three possibilities. This equation is based directly on known production capabilities.

An implicit (desired) capability for the ALCM framework is the entry of new relationships or the revision of presented CERs. It is envisioned that the ALCM become at least a common interface for accepted data and CERs applicable to the automation and robotics for space

systems. The construction of an all-encompassing cost model for this purpose is probably not realistic nor desirable. However, the use of the ALCM for capture and usage of pertinent data is desired. Section 7 contains a series of suggested improvements to the current ALCM framework prototype. In that section, discussion includes the relationship of the ALCM and other cost modelling resources. If the ALCM can be used as a starting point that is integrated into the design process of future space systems, then its objective will have been accomplished.

5.3.3. Development of Cost Estimating Rationale For Robotic Systems/Subsystems

For this phase of the ALCM development, development of preliminary cost estimating rationales support top-down/parametric estimates. The equations listed below are only preliminary, since their basis is two data points. These data relate to the information associated with the Tile Rewaterproofing Robotic System (Test Case 1) and a simple generic robotic system (Test Case 2). Rockwell's SSD Robotics Group provides all the data to the left of the 'SUM' column in both Tables 16 and 18. During a follow-on study, updating these equations reflects analysis of other robotic systems and their attendant costs. The purpose of the two tables was to determine the respective sigmas (σ) and lambdas (λ). Once determined, they apply to the Equations #1 and Equation #7 respectively.

Table 16 and Equations #1 through #6 in Table 17 associate with mobile base platforms for robotic systems. Equation #1 is the aggregate expression to estimate cost given the parameters for payload capability, positioning accuracy and type of control mechanism, and respective complexities and technology readiness factors. Equations #4 through #6 depict the mathematical expressions to account for cost impacts due to the three driving factors associated with mobile base platforms.

Test Case	F(x)		Unit	Complexity	Technology	Sum	PC1	PA1	CM1
1	\$70.0	Payload Capability (lbs)	100	1	6	1.2648	0.5385	0.2001	0.5220
		Positioning Accuracy (in.)	0.5	2	6				
		Control Type	2	2	6				
2	\$2000.0	Payload Capability (lbs)	2500	3	6	\$1034	1.7081	0.3452	1.0800
		Positioning Accuracy (in.)	0.05	2	6				
		Control Type	3	1	4				
		Payload Capability (lbs)	150	1	6	\$1285	0.5782	0.4508	2.1000
		Positioning Accuracy (in.)	0.1	2	5				
		Control Type	3	3	4				
Sigma	-1.0883			L (lambda)					
Lambda	18.10884359								
KVAR1	0.45								
KVAR2	0.2								
KVAR3	0.35								
KVAR4	1								
KVAR5									

Table 16 Spreadsheet Analysis of Mobile Base Platforms To Determine Calibration Coefficients.

$$\text{Equation \#1: } MBP\$\$ = \lambda \cdot e^{-\sigma \cdot ((PCL1 \times xvar1) + (PAL2 \times xvar2) + (CNL3 \times xvar3))}$$

$$\text{Equation \#2: } \lambda = 18.1068$$

$$\text{Equation \#3: } \sigma = -1.0693$$

$$\text{Equation \#4: } PCL1 = (\log_{10}(wt \times 2.5)) \times (CL + 1) \times \left(\frac{8 - TRL}{8}\right)$$

$$\text{Equation \#5: } PAL2 = \left(\log_{10}\left(\frac{10}{ATAN(PA)}\right)\right) \times (CL + 1) \times \left(\frac{8 - TRL}{8}\right)$$

$$\text{Equation \#6: } CNL3 = (CNTL_{Type}) \times (CL + 1) \times \left(\frac{8 - TRL}{8}\right)$$

Table 17. CER Generation is Highly Reliant Upon Experienced Personnel and Mathematical Expression Manipulation Tools.

Table 18 and Equations #7 through #14 in Table 19 associate with the manipulator subsystem components for robotic systems. Equation #7 is the aggregate expression to estimate cost given the following parameters:

- # of degrees of freedom
- reach capability
- payload lift capability
- positioning accuracy
- type of command and control mechanism
- respective complexities and technology readiness factors.

Equations #8 through #14 depict the mathematical expressions to account for cost impacts due to the five driving factors associated with manipulator subsystems.

Test Case	F(x)	Unit	Complexity	Technology	SUM	DOF1	RCH2	PLC3	PA4	CC5
1	\$1200	Degrees of freedom	4	2	8	0.7722	0.2000	0.1276	0.1274	0.0173
		Reach (ft)	10	3	8					
		Payload Lift Cap (lbs)	20	2	8					
		Positioning Accy (in)	0.05	2	8					
		Command/Control	1	2	8					
2	\$750.0	Degrees of freedom	4	1	8	1.1359	0.0570	0.0000	0.1610	0.0000
		Reach (ft)	10	2	8					
		Payload Lift Cap (lbs)	20	2	8					
		Positioning Accy (in)	0	0	0					
		Command/Control	1	1	8					
Sigma	-5.0398			Ln(lambda)						
Lambda	2.4488									
XVAR1	0.1									
XVAR2	0.1									
XVAR3	0.3									
XVAR4	0.1									
XVAR5	0.4									
				Yval						
				\$1531.975						

Table 18. Spreadsheet Analysis of Manipulator Systems To Determine Calibration Coefficients.

Equation #7: $MANSY\$\$ = \lambda \cdot e^{-\sigma \cdot (DOF1 \times xvar1) + (RCH2 \times xvar2) + (PLC3 \times xvar3) + (PA4 \times xvar4) + (CC5 \times xvar5)}$

Equation #8: $\lambda = 2.4488$

Equation #9: $\sigma = -5.0398$

Equation #10: $DOF1 = \left(\frac{DOF^2}{6} \right) \times (CL + 1) \times \left(\frac{8 - TRL}{8} \right)$

Equation #11: $RCH2 = \left(\frac{5000}{(RCH - 150)^2} \times 5 \right) \times (CL + 1) \times \left(\frac{8 - TRL}{8} \right)$

Equation #12: $PLC3 = \left(\frac{\log_{10}(wt \times 2.5)}{3} \right) \times (CL + 1) \times \left(\frac{8 - TRL}{8} \right)$

Equation #13: $PA4 = \left(\frac{\log_{10}\left(\frac{10}{ATANPAJ}\right)}{10} \right) \times (CL + 1) \times \left(\frac{8 - TRL}{8} \right)$

Equation #14: $CC5 = (C \& C_{Type}) \times (CL + 1) \times \left(\frac{8 - TRL}{8} \right)$

Table 19. The Manipulator System CER Uses all Associated Major Design Parameters.

Figures 8 through 10 contain the analyses for derivation of the CERs for reach, payload capability, and positioning accuracy. Inclusion of these additional charts corresponds to the relative complexity of the CER equation format.

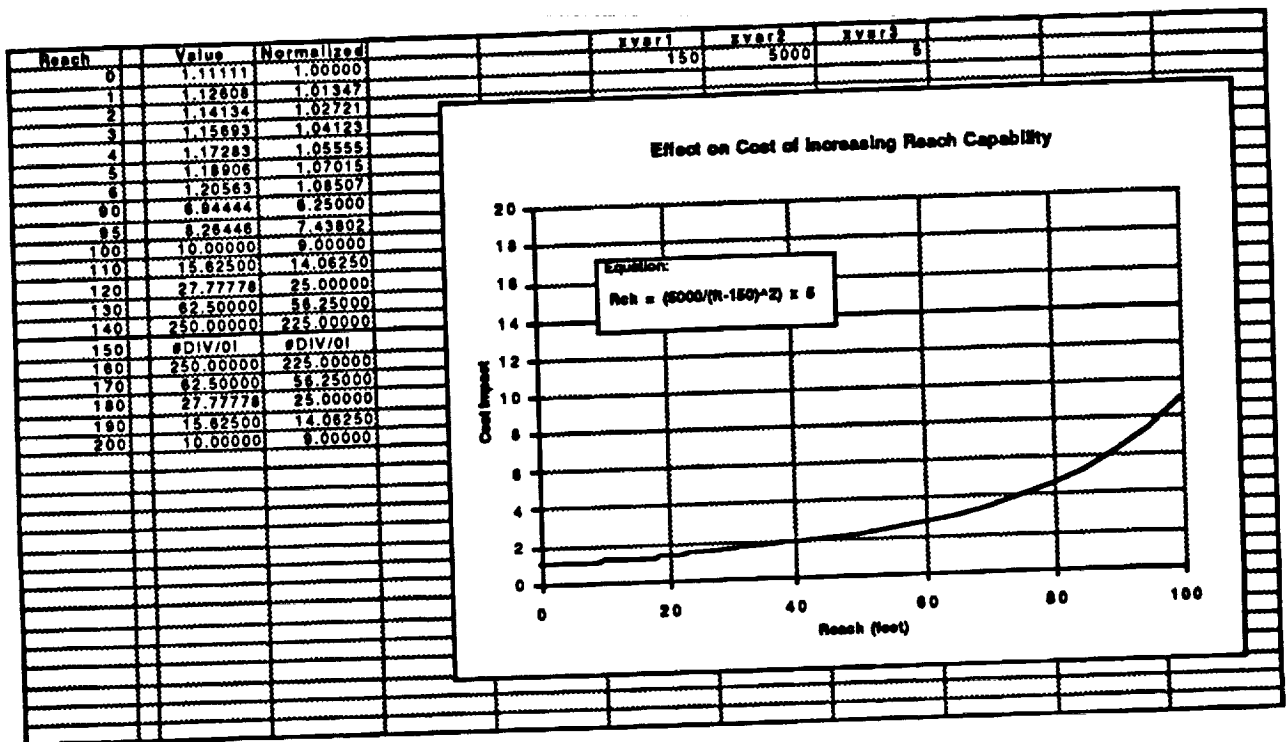


Figure 8. Reach-related Costs Rise Quickly Over 40 Feet.

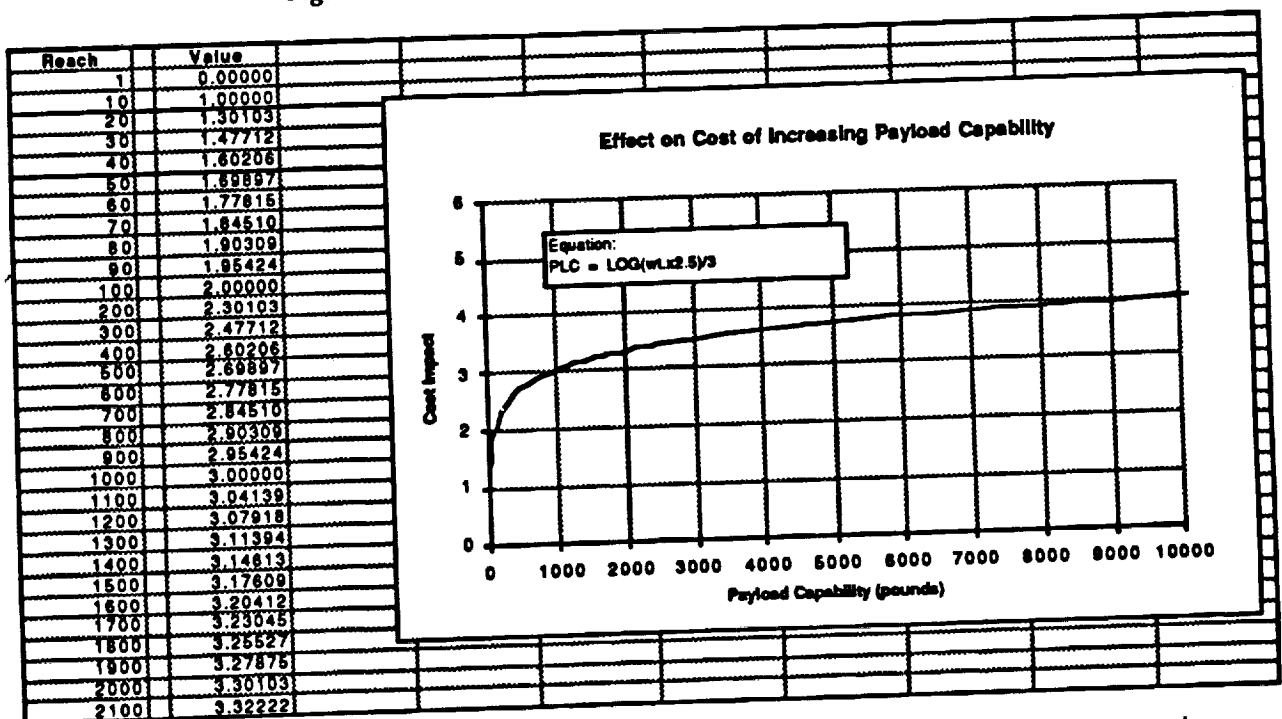


Figure 9. Payload Capability Cost is Very Dependent Upon Weight Until Around 200 Pounds.

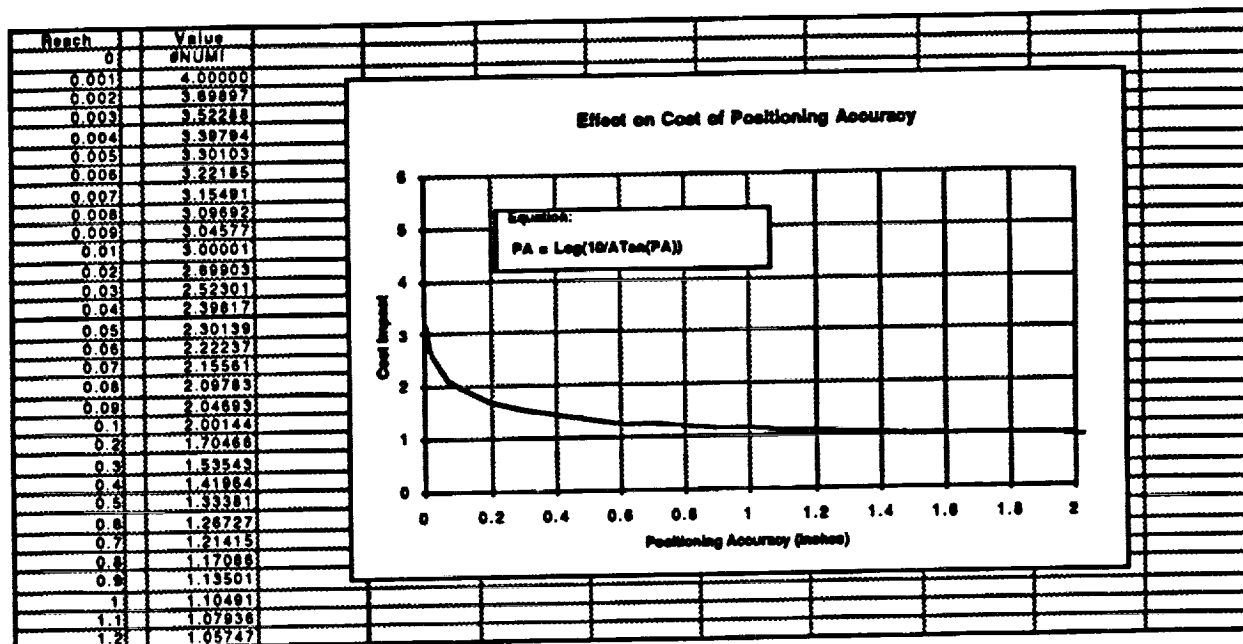


Figure 10. Cost Associated with Positioning Accuracy Rises Dramatically with Requirements Less than 0.1 Inch.

5.4. Framework Prototype

The framework prototype generated during this contractual effort is envisioned as a precursor to the ultimate cost modelling tool. This framework provides a proof-of-concept and user interface testbed to facilitate requirements' definition and tool utility. The evolutionary approach to the ultimate tool's construction benefits both the developer and user. Incremental releases of the software allow the true intentions of the user be incorporated into the software model. Discoveries of new or different requirements may occur throughout the life-cycle of the software. The user is not "stuck" with the delivered software at the end of the contract.

This section describes the capabilities of the framework, its implementation details, and provides a mapping to show how the prototype has met its objectives for this program.

5.4.1 Framework Overview

Section 3 describes the objectives for this program. They are decomposed into two main partitions: near-term and far-term. The framework prototype developed for this program responds to the near-term objectives with the far-term set accounted for in the envisioned growth plan for the software. Figure 11 (Figure 2 is repeated here for convenience) documents the near-term objectives demonstrated in the prototype.

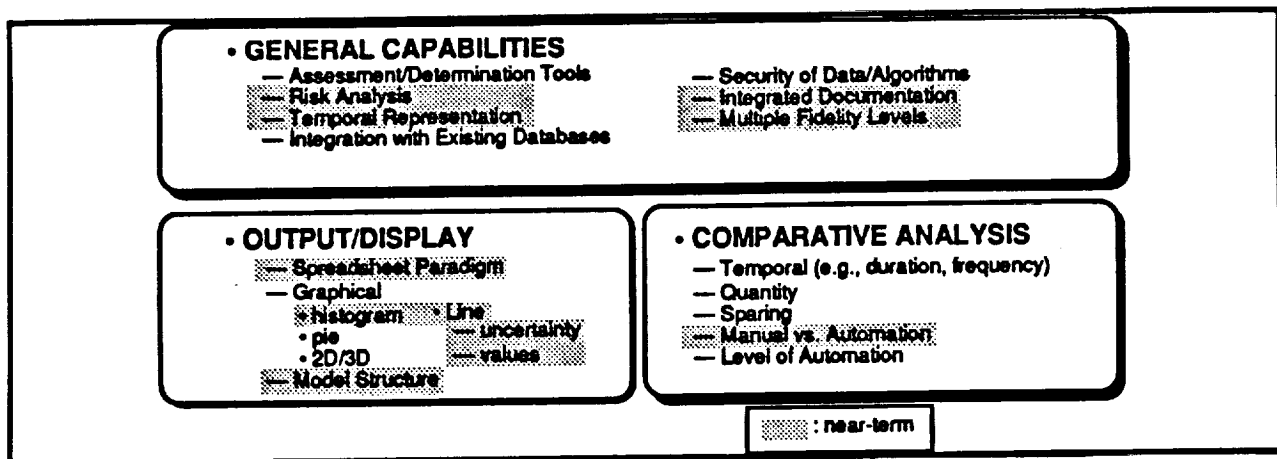


Figure 11. Near-term Objectives are an Achievable Subset of the Envisioned Capabilities of the Final ALCM Tool.

The capabilities of the framework prototype are categorized into 3 main areas: general capabilities, output/display capabilities, and comparative analysis. Figure 11 provides a graphical summary of the 3 categories. The Rockwell approach to capitalize upon the DEMOS development environment permits the categories to be actualized and therefore its features are subsumed by the other categories' capabilities. DEMOS allows the timely development of the framework's features but is not an intrinsic requirement for the ALCM framework. Its use reduces risk and cost associated with the sophisticated features of the model's implementation. Appendix 9.3 contains a graphical tutorial on the use of DEMOS.

The General Capabilities implemented in the framework prototype include:

- Risk analysis
- Temporal representation
- Integrated documentation
- Multiple fidelity levels

The prototype supports risk analysis in several ways. First, the ability to specify and model parameters in the problem space that may contribute to the risk of the solution. Second, decision analysis mechanisms, provided by DEMOS, support the association of uncertainty to objects within the model. Uncertain variables are not imposed by DEMOS. However, the variables chosen to be probabilistic in the prototype include cost, technology maturity, and sizing values. Lastly, sensitivity analysis may be performed to identify key drivers in the model.

Temporal representation is handled by the explicit modelling of the life-cycle phases. Costs may be allocated to the appropriate phase of the project in question. Costs may be evaluated at two levels of granularity, the 3 main phases and their decomposition (refer to Figure 5 for the life-cycle phase definitions).

The prototype includes documentation for each object (variable). Selection of an object permits the review of a suite of information including variable type, units, title, description, definition, inputs, and outputs. The variable type indicates the potential usage of that object within the model. Options include index, decision, change, submodel, and criterion. Refer to Appendix 9.3 for a description of each of these types and their utility within the DEMOS environment.

The prototype is constructed in a hierarchical manner to permit varying levels of fidelity to be modelled within the same environment. Use of DEMOS submodels supports this capability. The framework prototype has been encoded with more detailed information for the automated option than the manual option due to the accessibility of data. The prototype, however, is oblivious to the degrees of fidelity and permits such mismatches in data quantity. This capability permits modelling situations without full knowledge to all levels of detail. This prototype allows, and encourages, the incremental development of a model.

The Output/Display Capabilities implemented in the framework prototype include:

- Spreadsheet paradigm
- Graphical display
- Model structure

The spreadsheet paradigm capitalizes upon human cognition capabilities to quickly discern complex relationships among data when they are organized into a matrix format. Scanning the horizontal and vertical indices accelerates the browsing capabilities and permits the precise identification of relationships among disparate data. The prototype uses this paradigm in several situations that share the common goal of projecting complex relationships in a compact and readily understandable format. Examples include the breakdown of cost by module and life-cycle phase and costs associated by WBS element and cost category. The DEMOS environment permits the display and editing of information in the spreadsheet format. Deterministic and probabilistic variables can reside in different "cells" within the same matrix.

Output and display capabilities are crucial to the utility of the prototype. Since a strength of the Rockwell approach is the iterative construction of the cost model, the degree to which it can be interacted with may be the measure of its success. The graphical representation of data can promote understanding by the user by an order of magnitude faster than a purely textual display [5]. The DEMOS environment, coupled with proven and standardized user interface protocols on the Macintosh platform, provides graphical display of numeric information in a variety of formats. Histogram and line charts present derived data in compact form. Also, multiple options can be presented within a single graphic. Trade-off analyses are facilitated by this "side by side" capability. The user is able to switch between histogram and line chart preferences throughout the model.

Model structure is presented graphically to the user through two main mechanisms: influence diagrams and hierarchical model construction. Influence diagrams provide a standardized charting methodology to define and manipulate models (refer to Figure 12). Explicitly displaying relationships among variables satisfies the qualitative nature of the problem. Influence diagrams also have a precise mathematical description that is the basis for much of DEMOS' power. Hierarchical model construction has been added to the classic influence diagram to permit the further decomposition of model elements. Bold lines graphically highlight submodel definitions. The coupling of integrated documentation with the graphical depiction of models permits the user to customize review (and construction) of the model to his or her own needs yet maintains a consistent and elegant definition of the problem domain. Refer to Appendix 9.3 for more detailed information.

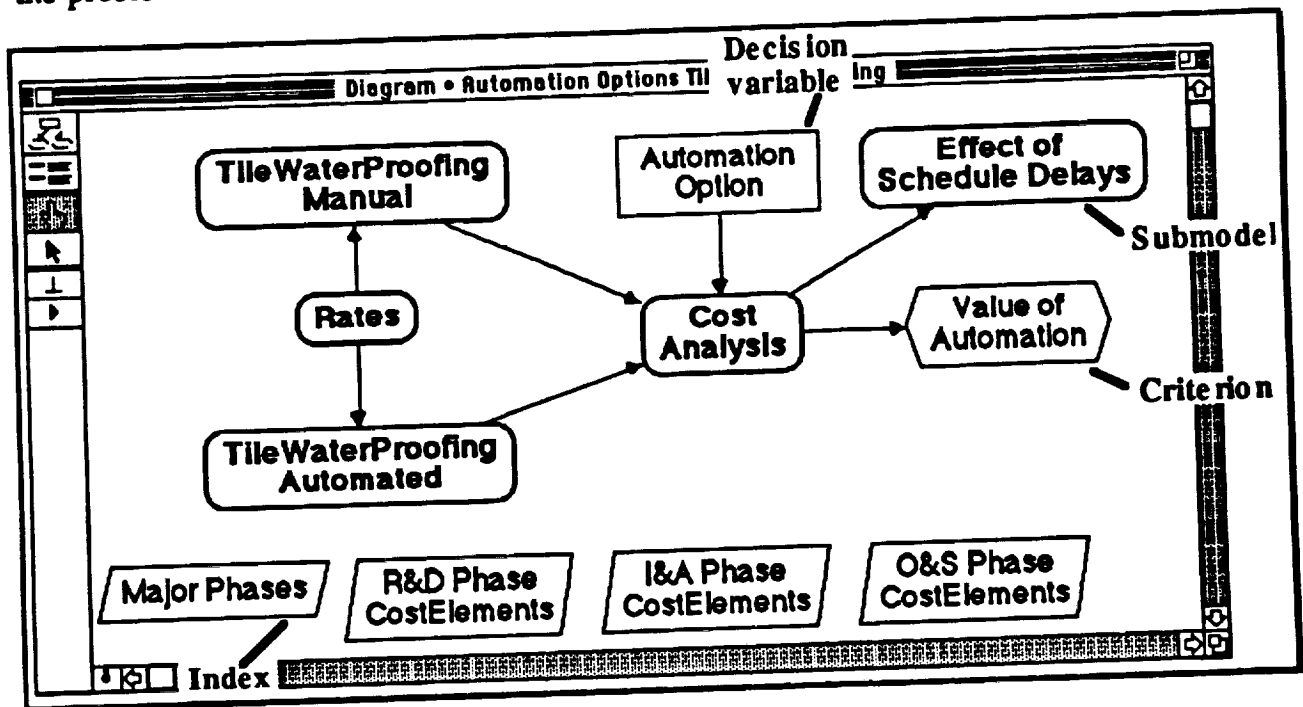


Figure 12. Influence Diagrams Are Used to Construct Sophisticated Models in a Readily Understandable Format.

Comparative analysis capabilities of the prototype permit the explicit modelling of manual versus automated options. The ability (and preference) to establish unique WBSs for each option is a feature. Experience has shown that manual and automated solutions to the same problem do not require the same hardware and software elements. The ability to explicitly model these differences in a cohesive manner is a strength of the framework. Comparison of the options can be viewed by phase and by a bottom-line figure dependent upon the user's needs and requirements.

5.4.2 Major Components

The ALCM framework prototype consists of several major components that address cost modelling over the life-cycle of a space system. A separable portion of the framework addresses the software elements of a system using an industry-accepted methodology. The major components have been structured such that the software cost elements are explicitly noted and integration of the two portions is eased. Also, the use of one portion of the framework does not require the other. However, their use in concert greatly expands the modelling capabilities of the system.

The major components include:

- Phase definitions
- Rate breakdown
- Program-specific (Work Breakdown Structures) WBSs
- Cost analysis features
- Automation bottom-line analysis

The software cost modelling portion is an add-on extension to the framework and is discussed in its own subsection.

Figure 13 portrays the top-level of the framework prototype within the DEMOS environment.

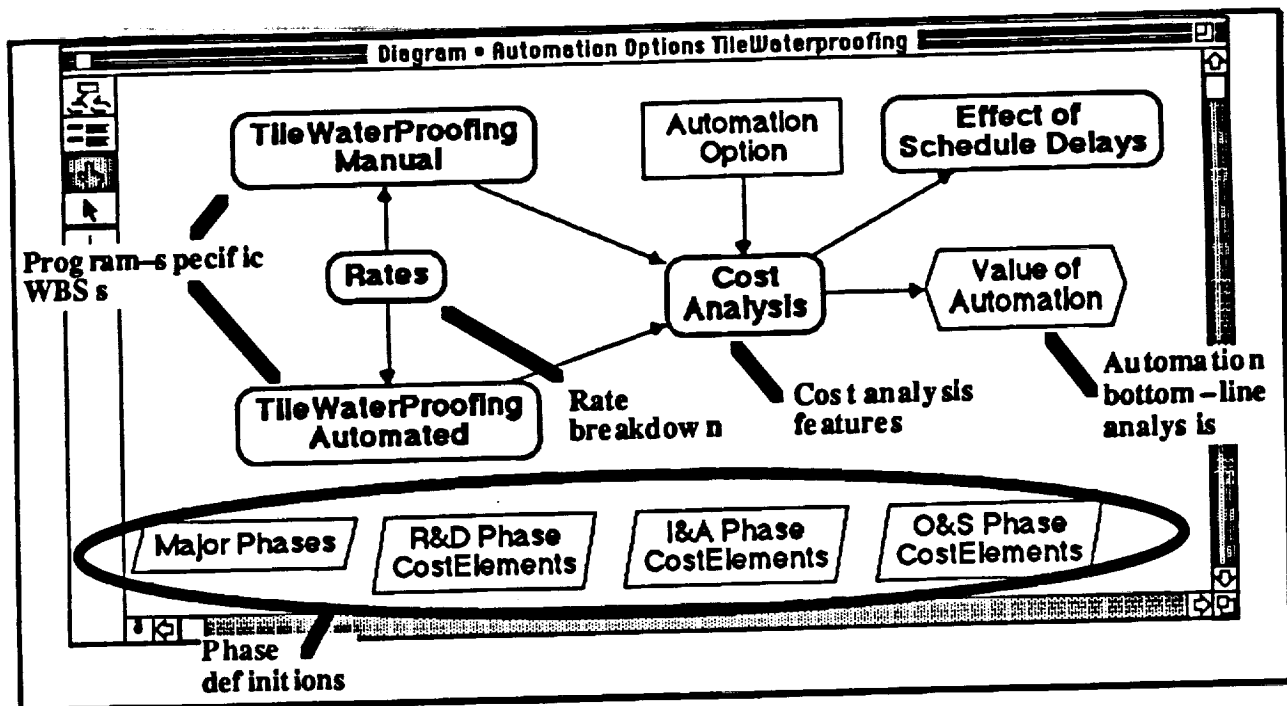


Figure 13. The Top Level of the Prototype Contains the Major Model Components for Ease of Use and Navigation.

5.4.2.1 Phase Definitions

As noted in Figure 13, the life-cycle phase definitions include two levels of fidelity: first, at the major phase level and, second, at the cost category level. This capability permits the user to review and evaluate information at the appropriate level of detail for his or her requirements. The major phases include Research and Development (R&D), Investment and Acquisition (I&A), and Operation and Support (O&S). Tables 13-15 depict the cost category definitions per phase. Figures 14-16 show the prototype's implementation for each phase's decomposition.

Object - R&D Phase CostElements

☐ Index Rd_costalts Units:

Title: R&D Phase CostElements

Description: Detailed cost elements for this Research & Development phase, standard for all projects.
 R&D phase includes Definition (Phase 0), Demonstration & Validation (Dem/Val) and Engineering & Manufacturing Development phases of RDT&E.

Definition:

'Development Eng (labor)'
'Producibility Eng&Plan(labor)'
'Tooling (capital)'
'Prototype Manufacturing (capital)'
'Data (labor)'
'System Test & Eval (labor)'
'System/Proj Mgmt (labor)'
'Training (labor)'
'Facilities (capital)'
'Software (labor)'
'Other (labor)'

Outputs: R&D% of EE-TFU

Figure 14. The Research and Development Cost Categories are Implemented as a List Which is Capable of Being Edited.

Object - I&A Phase CostElements

Index la_costelts Units:

Title: I&A Phase CostElements

Description: The cost categories included in this investment and acquisition phase of the life cycle. These costs include production non-recurring and recurring costs, initial deployment, and other initial investment.

Definition:

'Non-recurring (capital)'
'Production (capital)'
'Eng Changes (capital)'
'Syst T&E (capital)'
'Data (capital)'
'System/Project Mgmt (labor)'
'Oper/Site Activation (capital)'
'Training (capital)'
'Initial Spares & Repair (capital)'
'Transportation (capital)'
'Facilities (capital)'
'Other (capital)'

Outputs: I&A% of EE-TFU

Figure 15. Investment and Acquisition Costs Include Non-recurring Elements Which Can Be Tailored Per Project.

Object - O&S Phase CostElements

Index Oa_costelts Units:

Title: **O&S Phase CostElements**

Description: Costs beginning at initial operating capability (IOC) and continuing thru disposal of the unit. Cost categories apply to launch o&s, orbit operations, mission o&s, de-orbit operation, and landing operations and support and disposal.

Definition:

'Personnel (labor)'
'Mission O&S (labor)'
'Orbit O&S (labor)'
'Deorbit Operns (labor)'
'Landing O&S (labor)'
'Maintenance-Labor (labor)'
'Maintenance-Material (capital)'
'Maintenance-Transportation (capital)'
'Spare (capital)'
'Training (labor)'
'Documentation (labor)'
'Support & Test Equip (capital)'
'Software Maint (labor)'
'Disposal (capital)'
'Other (labor)'

Outputs: **O&S Phase Costs EE**

Figure 16. Operation and Support Costs Typically Account for the Bulk of Life-cycle Cost and Can Be Reviewed in Detail.

These elements are coded as lists that can be easily revised. These lists are indices in the cost analysis portion of the model to allow review of cost by phase and cost category.

5.4.2.2 Rate Breakdown

Rates define the relationship between elements and labor, and the system's cost. This factor is highly variable and dependent upon several mitigating factors that include year of execution, type of accounting method, and even geographic location of the work being performed. Further complicating the situation is that a life-cycle of a space system is typically expressed in years. Inflation affects the dollars in a particular year when compared to those spent in another.

The ALCM framework accounts for this volatility of rates within the cost modelling domain by dedicating a submodel to the rate breakdown. This submodel is also responsible for converting between those cost elements that are best expressed in dollar amounts to those efforts expressed in hours. The framework relates the elements in the dollar domain.

Four main element types comprise the rate breakdown (refer to Figure 17):

- Capital rate—used to specify those elements best described in dollar amounts
- Labor rate—used to specify those elements best described in hour amounts
- Rate vectors—conversion between dollar and hour amounts
- Inflation factors—time-phasing of money to account for using standard-issue inflation factor tables

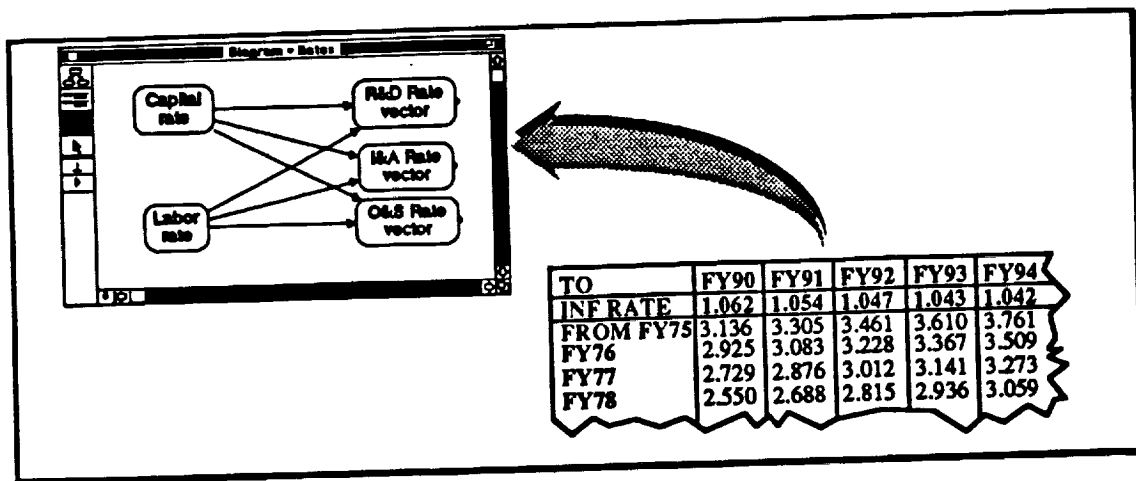


Figure 17. Time-dependent Factors of Cost are Modelled Through the Rate Breakdown Submodel.

The submodel expresses cost elements in a common domain (i.e., dollars) and permits time-phased spending to be view from an agreed upon vantage point (i.e., fiscal year). This capability also supports modelling costs when elements of a program slip in time. Ramifications of schedule slippage will be a future capability of the ALCM, but its essential elements are present in the current framework.

5.4.2.3 Program-specific WBSs

Implementation of a system using manual or automated techniques frequently relies upon a different set of hardware and software components. The WBS is a mechanism used to organize cost and work information to ensure consistency and increase management visibility. Accordingly, the WBS of each option will differ. The framework prototype supports this reality by allocating a submodel to each potential implementation. The WBS contained in each submodel is unique for that manual or automated system.

The ALCM framework supports multiple cost estimation methodologies. These include, but are not limited to, bottom-up (accounting and parametric) and top-down (CER-based) (refer to Figure 18). Additionally, capabilities are integrated to examine the various methodology results. Specifically, the current version of the framework compares bottom-up and top-down methods.

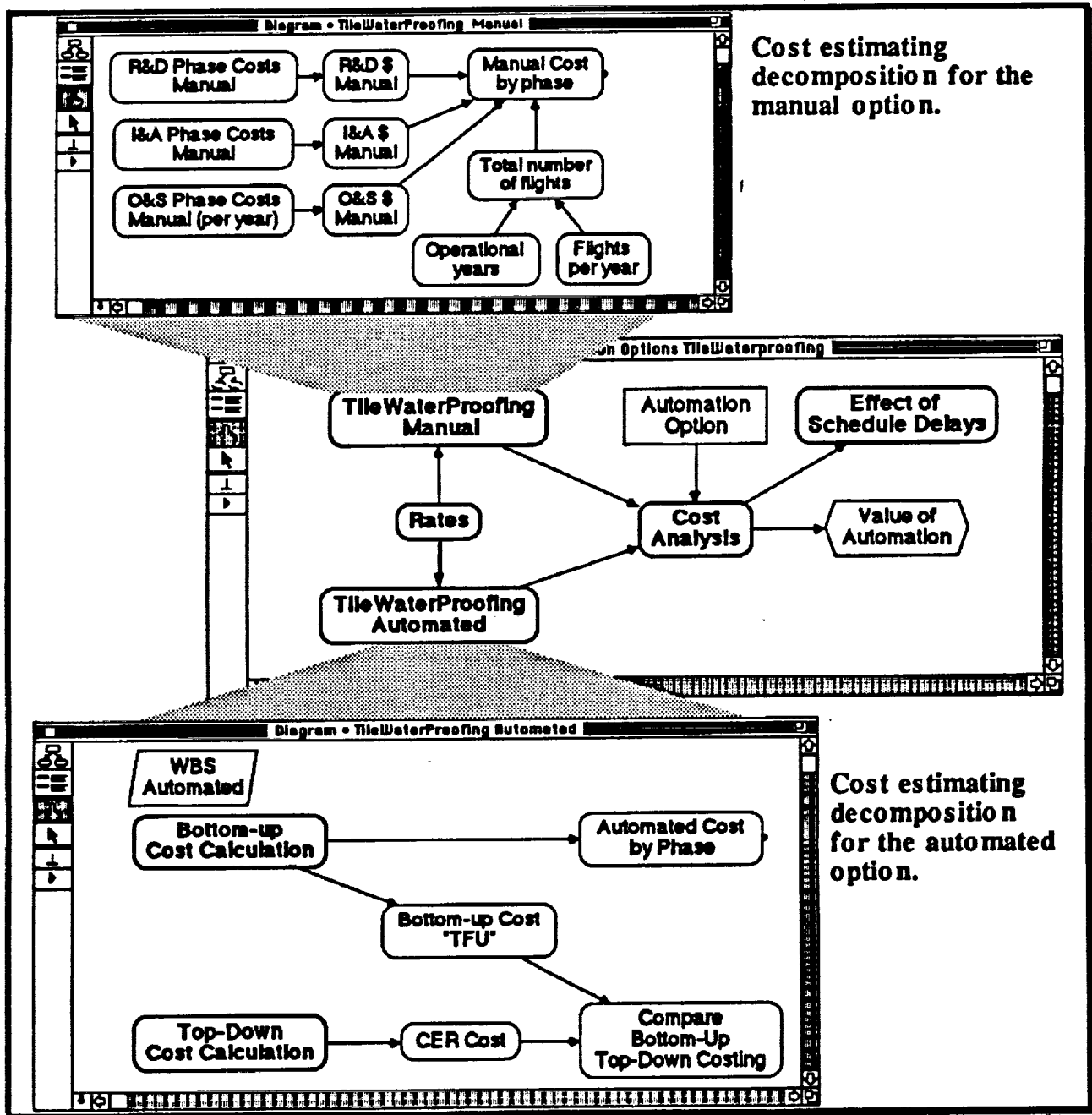


Figure 18. The ALCM Framework Supports Multiple Cost Estimation Methodologies and Their Comparative Analysis.

In particular, Figure 19 shows the various ALCM framework cost estimating capabilities. The manual option uses bottom-up estimation with accounting data. The automated option uses both accounting data and parametric relationships for bottom-up costing and CERs for top-down estimation.

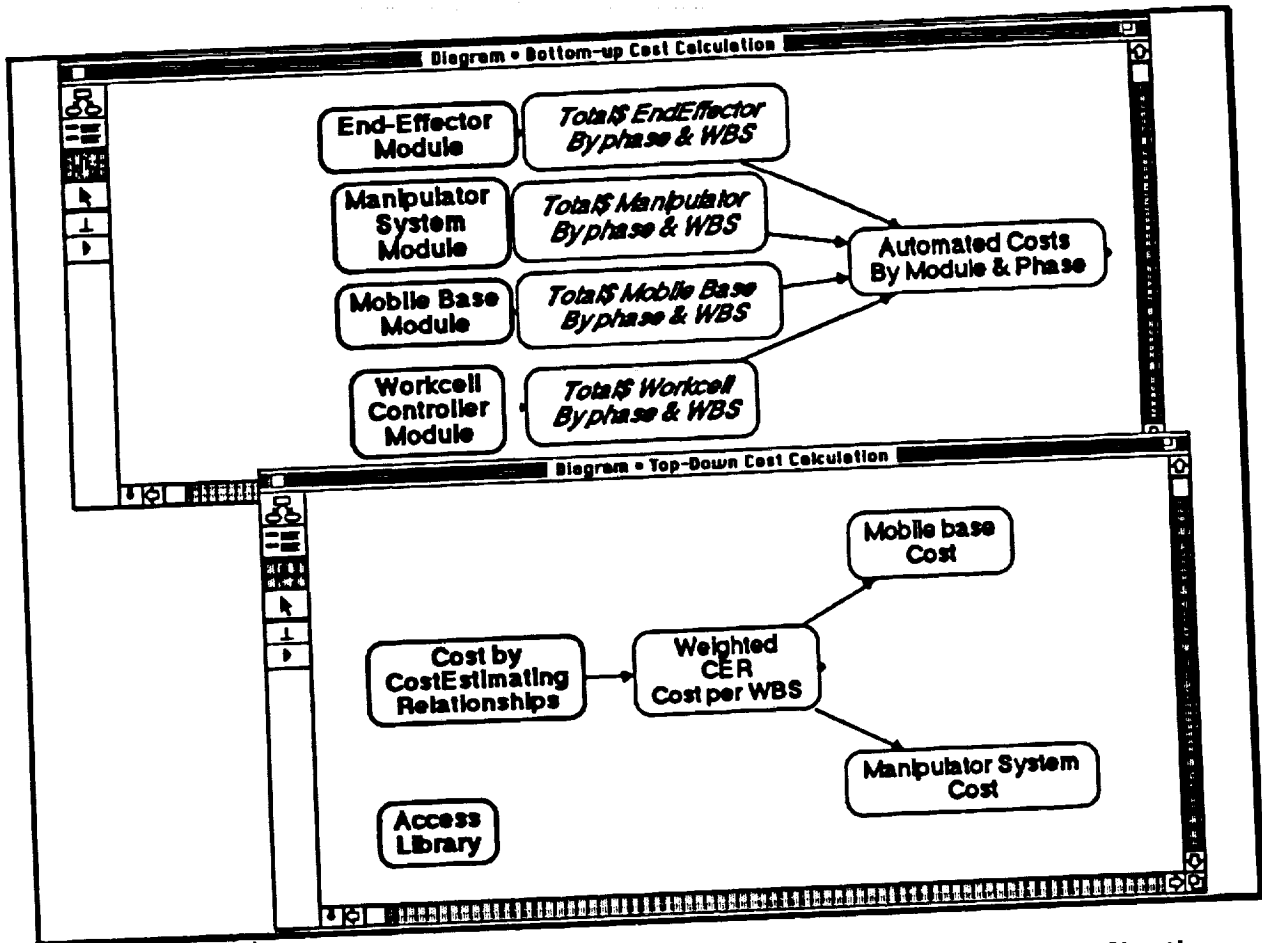


Figure 19. Different Estimation Methodologies Are Applied to Appropriate Costing Situations.

The program-specific WBSs represent the heart of the prototype. It is in these structures that the costs, estimates, and relationships are expressed and utilized. The subsystem components comprise the WBS for each implementation. The user and the availability of data determine the amount of detail entered. The DEMOS environment provides the mechanism for engineering "guesstimates" for inclusion in the model along side firm vendors quotes and validated cost estimating relationships. Figure 20 provides a pictorial view of the capabilities of the prototype to define a WBS, describe the individual components, express cost data associated with a component, and graphically review the costs for each component.

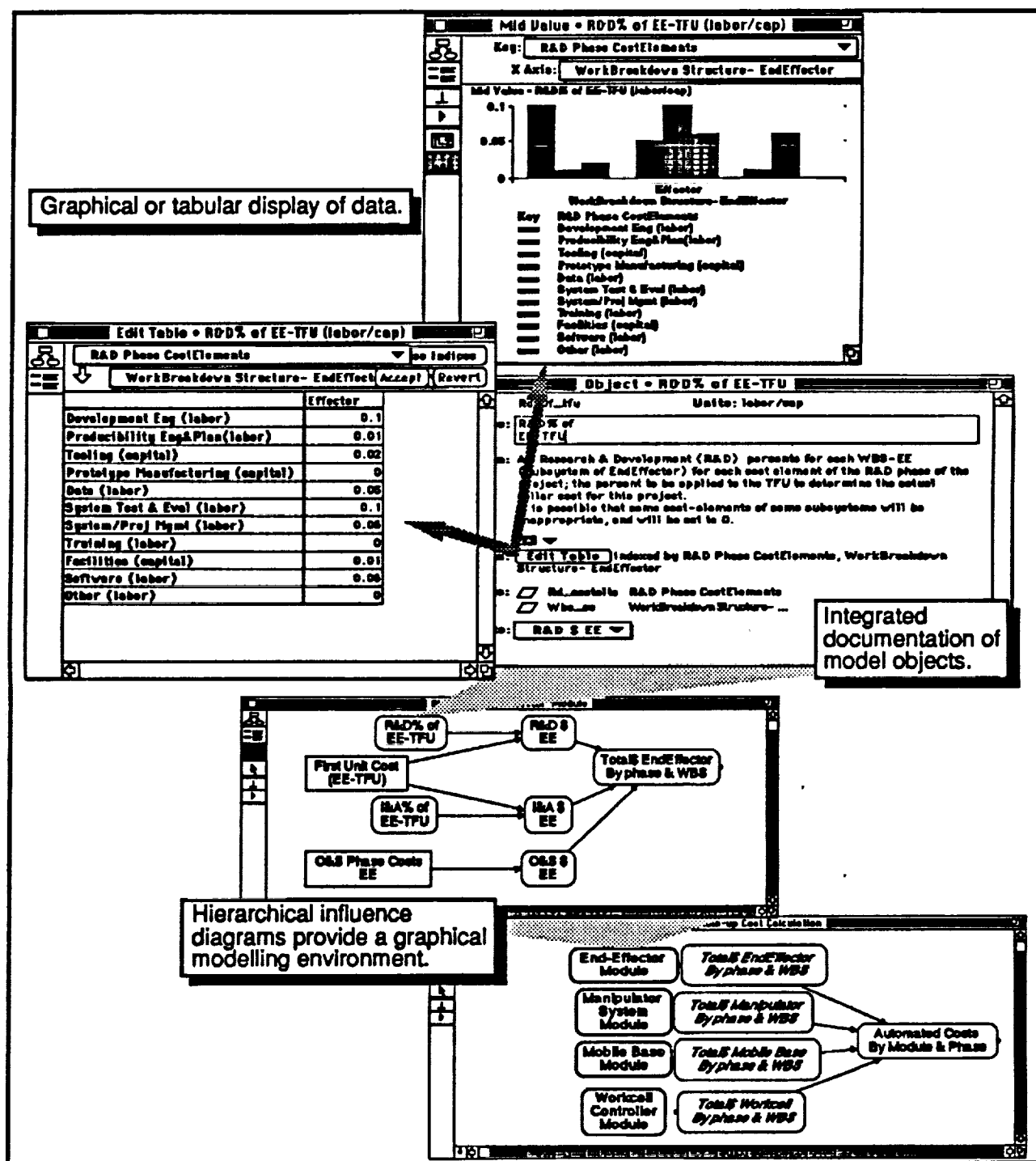


Figure 20. The ALCM Framework Prototype Provides a Flexible Environment for Constructing and Reviewing Space System Cost Models Based on Work Breakdown Structures.

5.4.2.4 Cost Analysis Features

The cost analysis features provide mechanisms to compare the manual and automated implementation methods of space systems. Costs may be examined by method and by phase.

Figure 21 provides the graphical screens of the prototype showing the cost breakdown by phase and method. Histograms depict cost by phase for each option while line chart shows the uncertain values associated with the methods. Histograms permit the user to quickly ascertain the key drivers of cost for the various implementations. Line charts present uncertain variables in those areas where firm numbers are not known, required, or available.

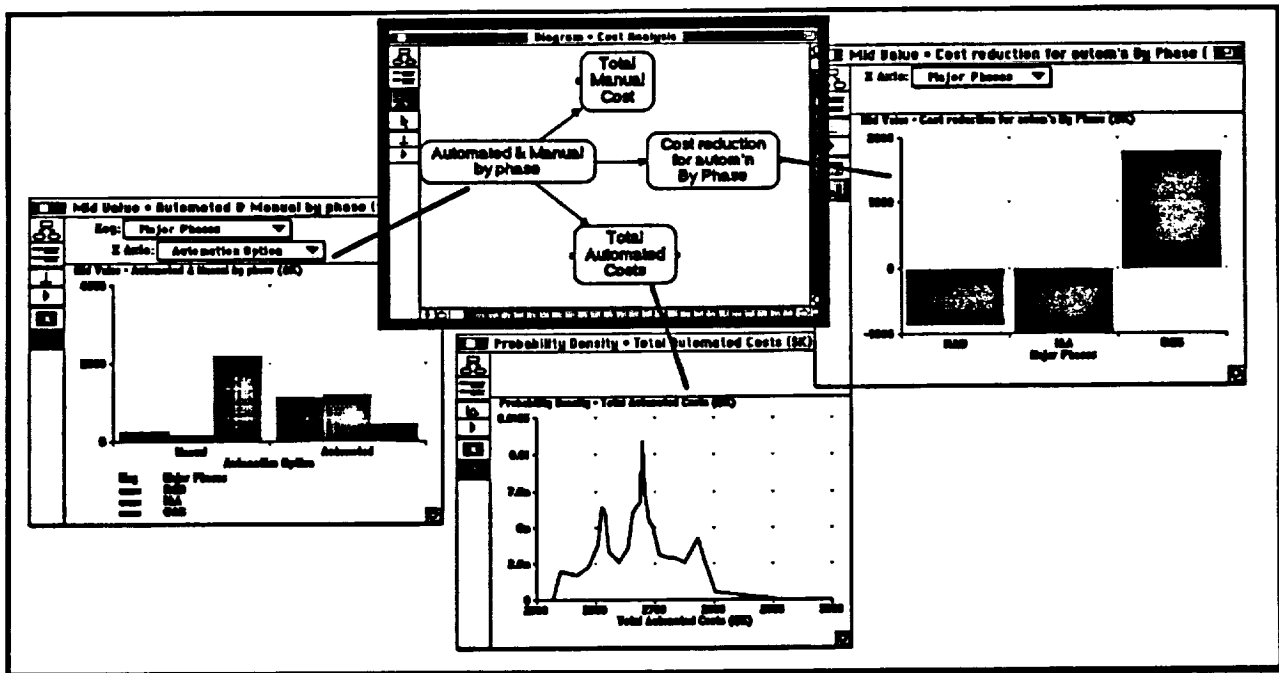


Figure 21. ALCM Framework Provides Cost Analysis Capabilities to Review Cost by Phase and Element to Determine Key Drivers in the Life-cycle.

5.4.2.5 Automation Bottom-line Analysis

The ALCM framework prototype provides a quick-look capability to examine the relative cost benefit from a manual or automated implementation. This single measure-of-merit may have an uncertainty associated with it based on the composition of the underlying model parameters. This analysis allows review of the bottom-line for the potential implementation options without being weighed down in the intricacies of the model. Of course, the user may examine the model to all its levels of details if one wishes. However, detailed perusal of the model is not required to obtain the answer generated by the model. Figure 22 portrays the information available from this quick look capability.

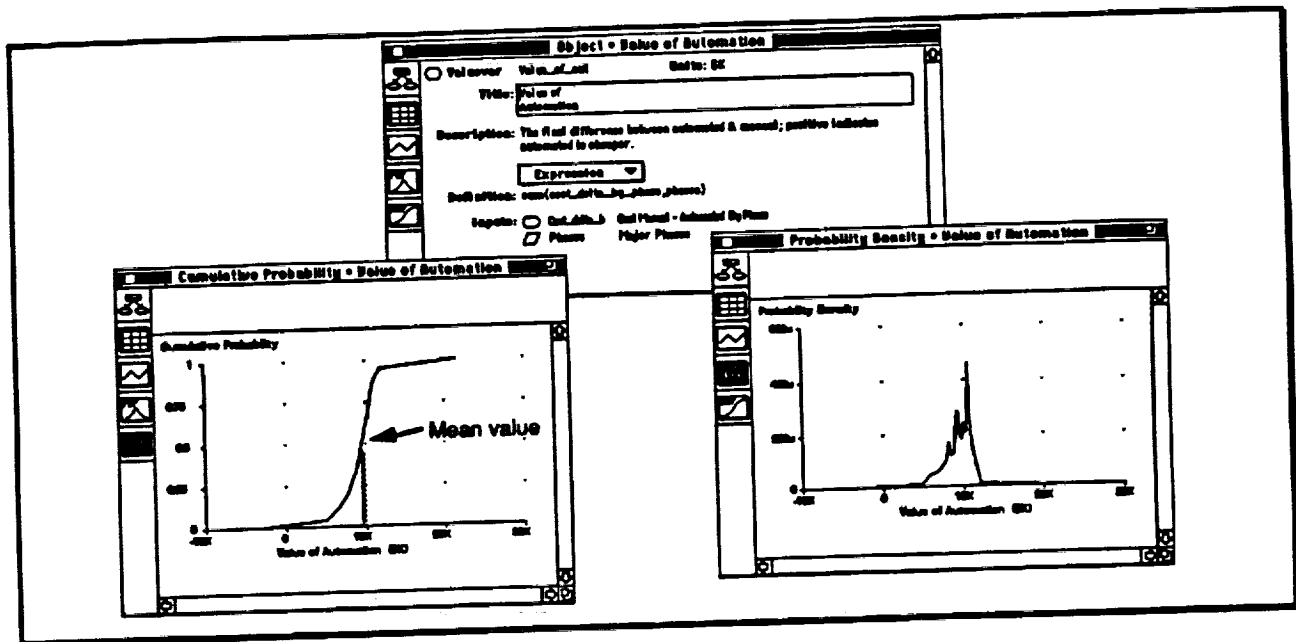


Figure 22. The Bottom-line Figure is Available From the Model Without Examining an Immense Amount of Detail.

5.4.2.6 Software Cost Modelling

Advances in processing hardware capabilities have accelerated the amount and extent of software included in space systems. This recent emphasis on software products within space systems has brought a new visibility to software and its life-cycle process. Software costs have dramatically risen in relative terms to the other elements of the system. As such, increased scrutiny of software-related costs and costing methods has occurred [8]. Software systems now require the use of a defined costing methodology (and an proven one at that).

Barry Boehm, former Director of Software Research and Technology at TRW, Inc., has been a forerunner in the field of software costing. His book, *Software Engineering Economics* [1], is considered required reading for software estimators. The cost estimation model described in that book, Constructive Cost Model (COCOMO), is a *de facto* standard within the aerospace community. This model is capable of generating cost, effort, and duration estimates for a described software system.

Using IR&D funds, Rockwell International's Space Systems Division has created a COCOMO model within the DEMOS environment entitled Autotrade. Autotrade is an embodiment of the intermediate COCOMO model as described in [1]. Autotrade also includes the following features not found within that standard model:

- Autonomy levels
- Probabilistic variables
- Sensitivity analyses

These additional parameters provide a unique flexibility to tailor the cost model to space systems in particular. The degree of sophistication for system capabilities and the efficiency of the software process affects the development and deployment of a system. Decision analysis permits real world considerations to be accounted for with probabilities and uncertainty associated with engineering estimates. Description and analysis of the software domain within decision analysis constructs allow detailed examination of the key drivers within the model and their relative relationships to other variables in the domain.

5.4.2.6.1 Autotrade Overview

Autotrade is a hierarchical software costing model built within the DEMOS environment. The top level consists of major definition and analysis components (refer to Figure 23):

- COCOMO cost drivers
- Size and productivity
- Autonomy levels definition
- CSCI identification
- Summary analysis

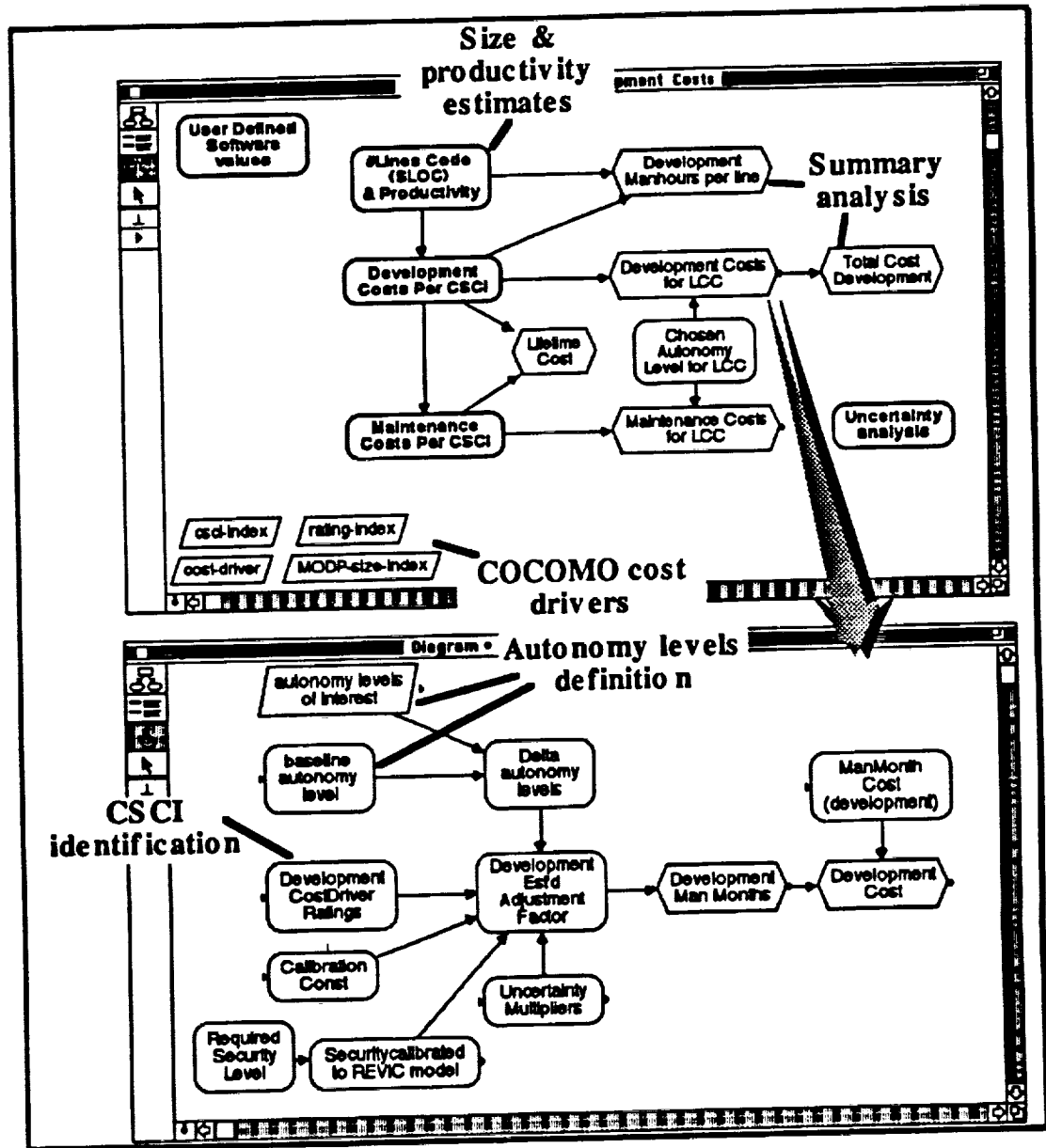


Figure 23. Autotrade is Functionally Partitioned to Ease Navigation of the Model.

The cost drivers element assigns the qualitative cost drivers to numeric values for calculations. The size and productivity portion permits specification of Computer Software Configuration Item (CSCI) size and the amount of reuse anticipated within the development and deployment of the software system. Productivity values are derived from these inputs. Autonomy levels per CSCI establish the sophistication of software capabilities at a system level. Allocation of qualitative values assigned to key cost factors used by COCOMO identify the CSCIs. The user observes composite cost and productivity measures from the top level.

5.4.2.6.2

COCOMO Cost Drivers

COCOMO's concept of operations is to assign qualitative values to several key factors. Each of these qualitative values has an associated numeric value. Mathematical combination of numeric values results in an overall adjustment factor for that CSCI. The adjustment factor is a multiplier in an empirically-derived curve fit for software costs. This portion of Autotrade relates the qualitative answers to their numeric counterparts. Figure 24 depicts the implementation and provides an identification for each of the factors.

- Qualitative values are assigned to each of the COCOMO factors (very low, low, nominal, high, very high, extra high)
- Factors Include:

- Required software reliability (RELY)
- Database size (DATA)
- Software product complexity (CPLX)
- Execution time constraint (TIME)
- Main storage constraint (STOR)
- Virtual machine volatility (VIRT)
- Computer turnaround time (TURN)
- Analyst capability (ACAP)
- Applications experience (AEXP)
- Programmer capability (PCAP)
- Virtual machine experience (VEXP)
- Programming language experience (LEXP)
- Modern programming practices (MODP)
- Use of software tools (TOOL)
- Schedule constraint (SCED)

- COCOMO factors encoded with a lookup table to relate qualitative and quantitative values

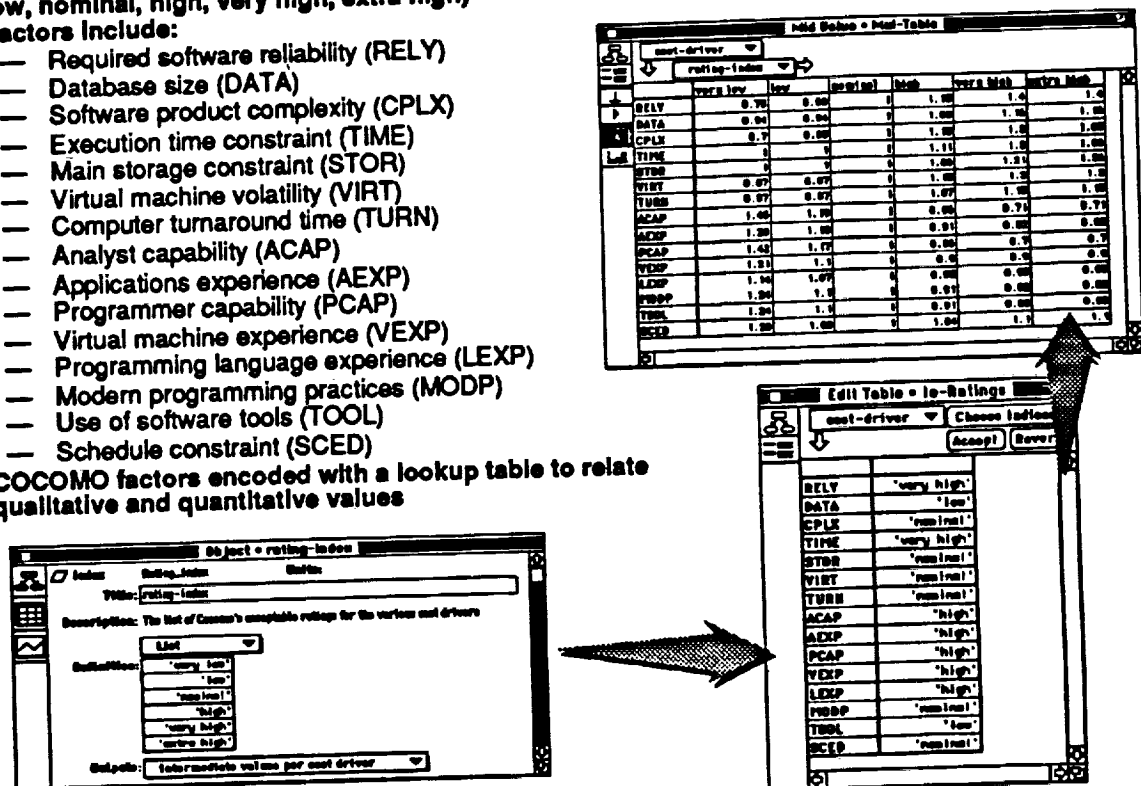


Figure 24. Autotrade Automatically Assigns the Correct Numeric Value to Each COCOMO Cost Driver to Provide Consistency.

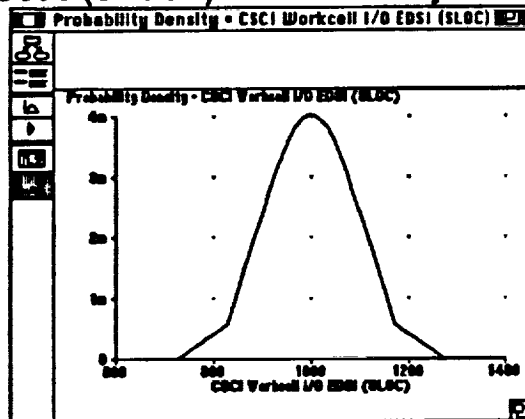
5.4.2.6.3

Size and Productivity

Autotrade provides the capability to specify CSCI size (in source lines of code) and parameters affecting productivity. The amount of reuse within the software life-cycle may be explicitly specified on 3 different levels: integration, design, and code. Each of these parameters affects the life-cycle uniquely. Combination of reuse percentages, using a heuristic relationship, provides a productivity measure for the CSCI. Figure 25 provides a pictorial view of CSCI size specification and the overall content of this component within

the model. Figure 26 shows various ways that derived productivity parameters may be presented to the user.

- Source Lines of Code (SLOCS) estimates may be expressed as probability distributions



- Amount of reuse is specified on 3 levels
 - Integration
 - Design
 - Code
- Reuse percentages are combined using a heuristic relationship
- Reuse and SLOCS are specified on a CSCI level

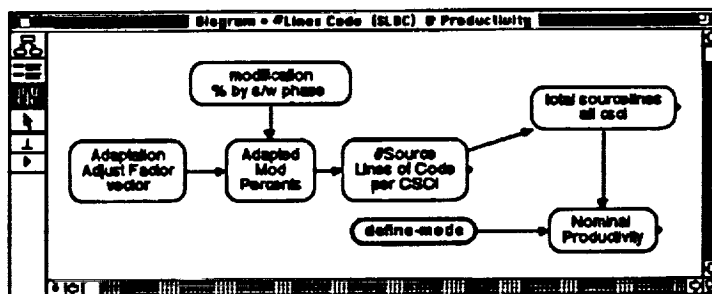
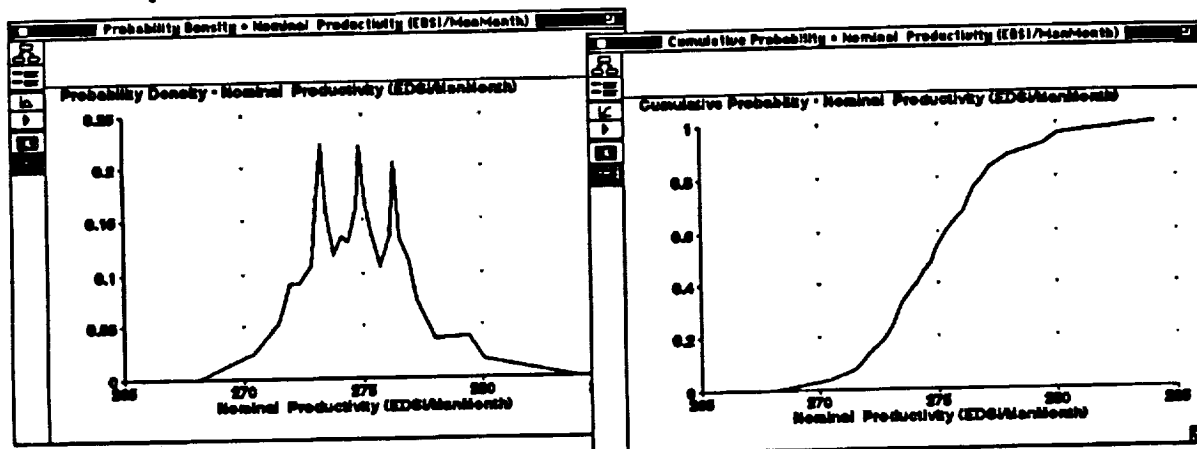


Figure 25. Autotrade Permits Uncertain Size Estimates for Each CSCI to be Used for Derivation of Productivity Parameters.

- Productivity values (distributions) are generated per CSCI and as a composite



- Total SLOCS (distributions) are generated per CSCI and as a composite

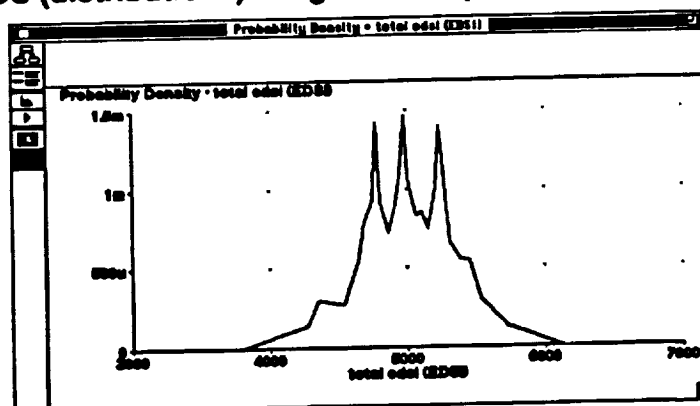


Figure 26. CSCI Productivity Measures May Be Analyzed From Several Perspectives.

5.4.2.6.4 Autonomy Levels Definition

Autonomy level definitions, derived under Rockwell IR&D funds, specify system capabilities to various degrees of sophistication [7]. These levels range from open-loop systems to totally self-sufficient systems. The levels, originally targeted toward spacecraft, have been extended to include ground segment functionality as well. Assignment of these capabilities is per CSCI. The specification of autonomy levels is a means for describing sophistication and complexity of the software domain for the system capabilities as a whole. These levels can be used to perform sensitivity analyses on cost versus sophistication. This capability encourages cost/benefit analysis for the software in a space system. Figure 27 depicts the graphical element in the model along with the summary autonomy level definition per ground segment function.

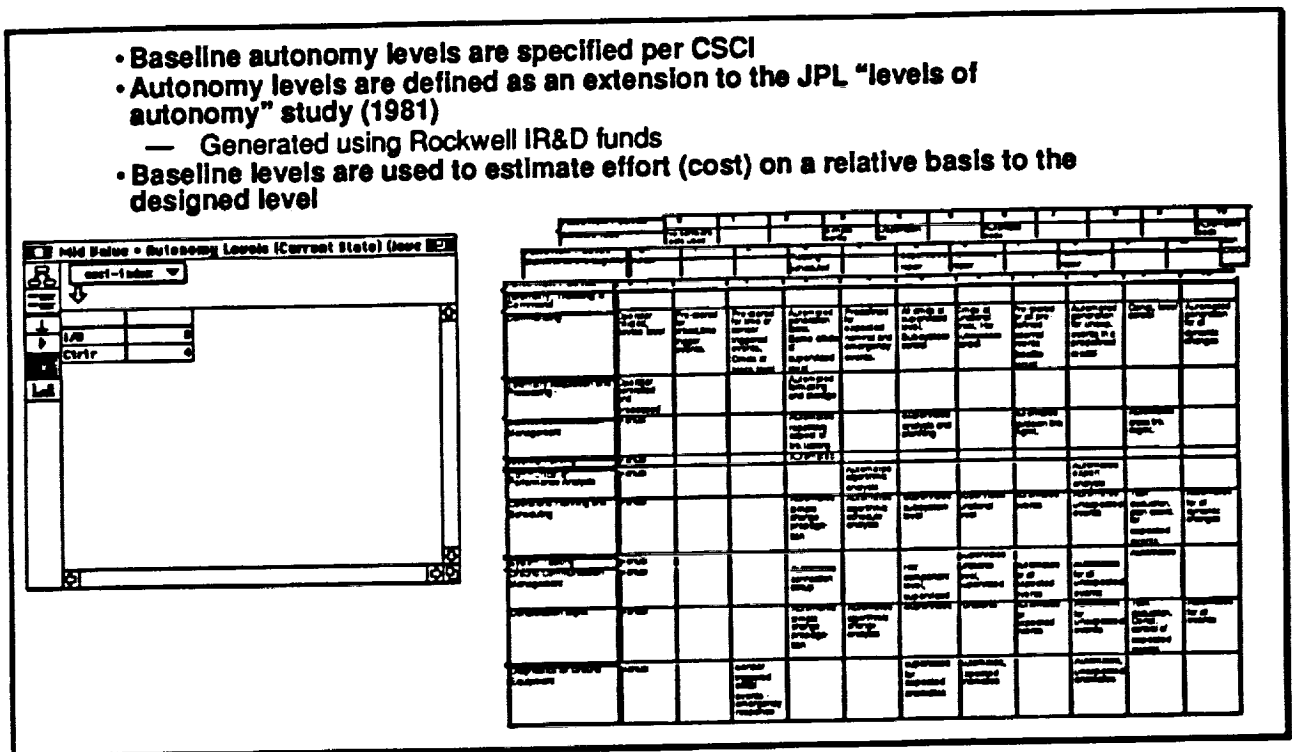


Figure 27. Specification of Autonomy Level per CSCI Supports Software Cost/Benefit Analysis for Space Systems.

5.4.2.6.5 CSCI Identification

CSCIs are uniquely identified and defined within Autotrade by assigning qualitative values to each of the COCOMO cost drivers and specifying the autonomy levels to be investigated. Figure 28 provides a graphical view of these capabilities. The cost driver values are entered as a qualitative value and Autotrade converts it to a numeric equivalent for use in COCOMO equations. Figure 29 shows the cost factors presented in a format that allows quick comparison of the relative values. COCOMO equations generate summary cost and duration values per each CSCI identified in the model.

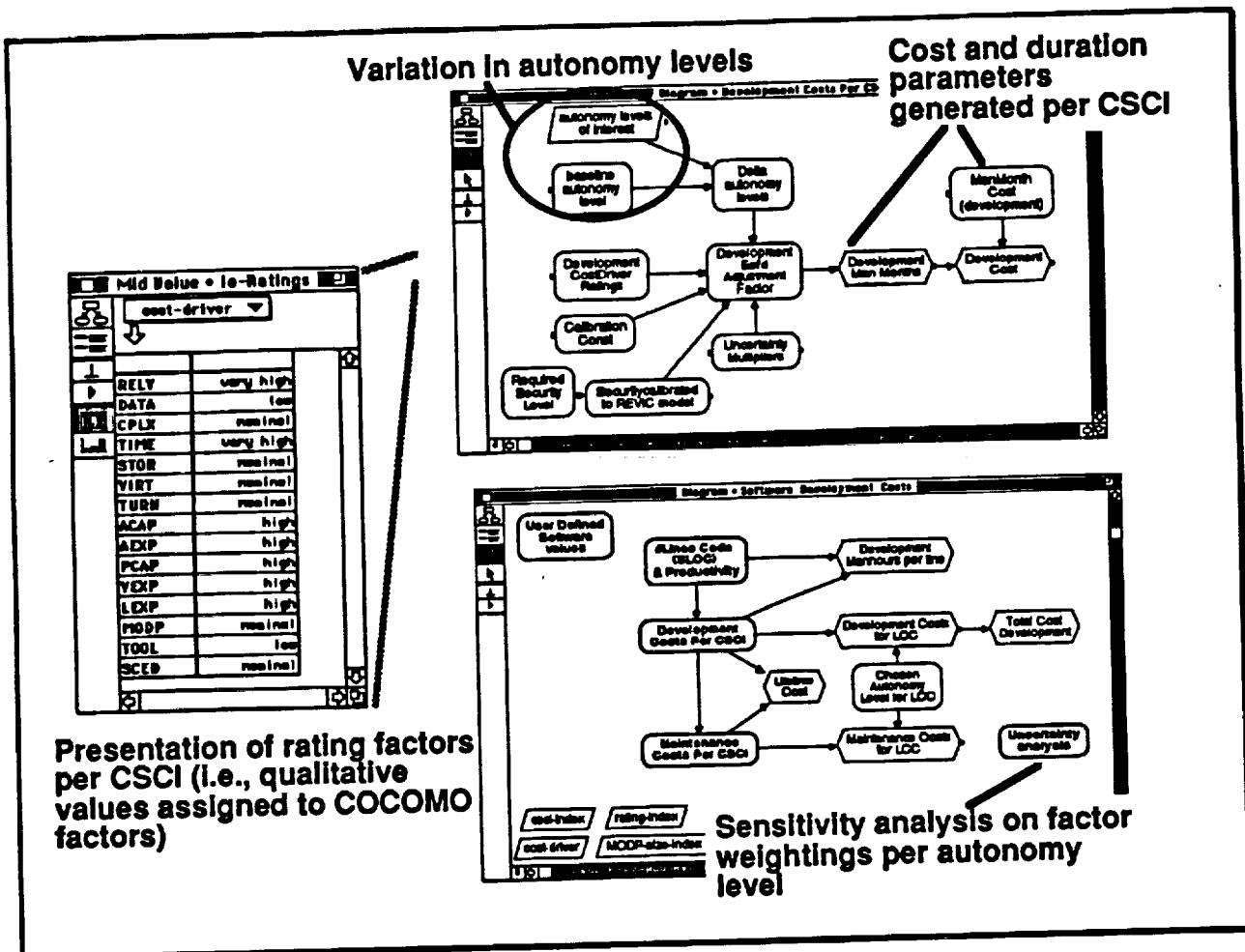


Figure 28. Cost Driver Values Are Specified For Each CSCI Permitting Detailed Analysis of the Envisioned System.

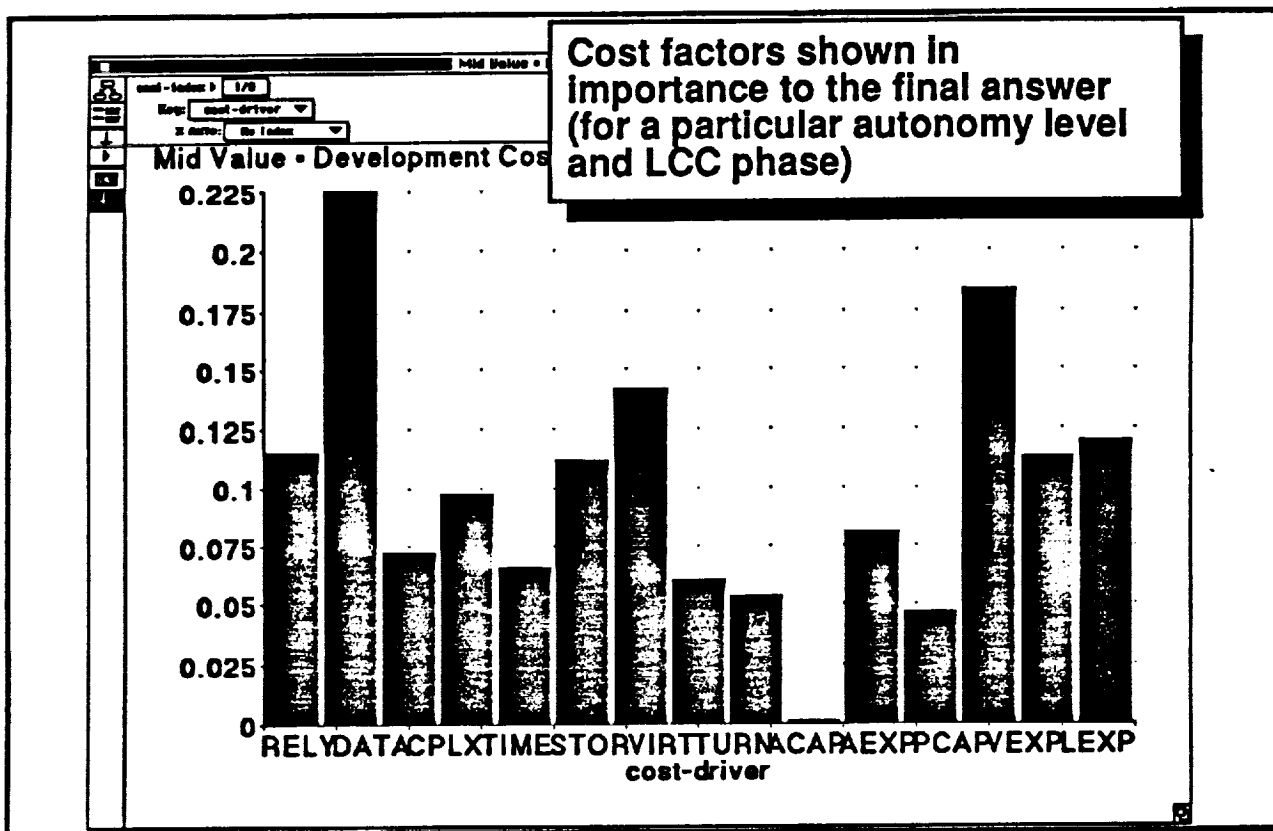


Figure 29. COCOMO Cost Factors May be Examined With Their Assigned Values per Autonomy Level and In Rank Order to Obtain Their Relative Importance.

5.4.2.6.6 Summary Analysis

Autotrade provides a summary analysis capability. Total software cost and productivity measures are presented for each level of sophistication envisioned by the user. This capability provides a top-level cost/benefit analysis without investigating the detail of the model. Of course, users are free to examine all levels of detail in the model at their leisure. The breadth of analysis can be manipulated from the top level, as well, by varying the autonomy levels to be investigated. Figure 30 provides a graphical representation of the summary analysis capabilities provided by Autotrade.

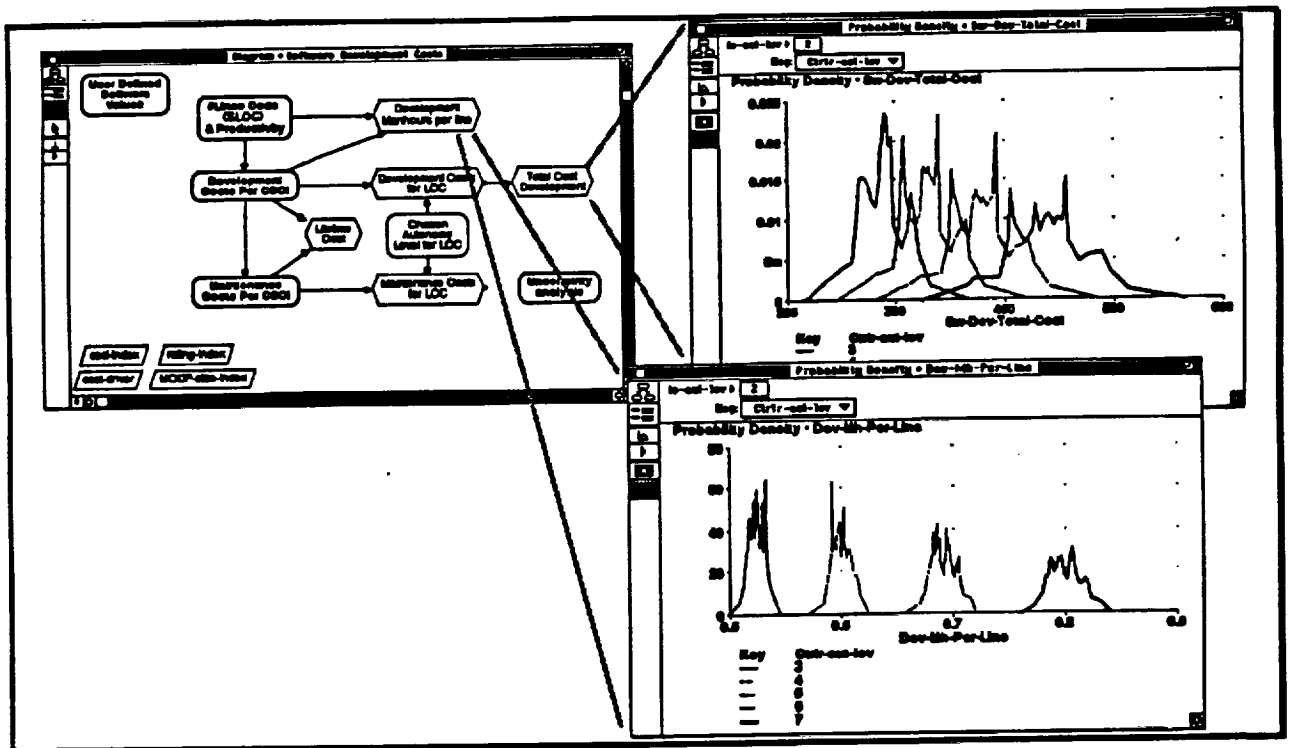


Figure 30. Summary Trade-off Analysis is Supported by Presenting Costs and Productivity Measures Over a Range of Autonomy Levels.

5.4.2.6.7 Autotrade Summary

Autotrade is a software cost environment, developed using Rockwell IR&D funds, that generates cost, duration, and productivity measures based on the COCOMO model. It expands upon the industry *de facto* model by adding the elements of autonomy levels, probability and sensitivity analysis. It is a highly interactive environment capable of handling uncertain input values and examining a range of sophistication levels. These capabilities encourage its use early in programs where there is a high degree of uncertainty within the problem domain both for estimates and system capabilities.

Autotrade is a valuable stand-alone model for costing space system software for ground and space segments. The ALCM framework allows it as an add-on module for sophisticated software costing of an entire system life-cycle. Its interactive environment is the same as the ALCM framework itself. As such, it shares the benefits of ALCM: ease of use, interactive dialogs promoting iterative refinement of cost estimates, and graphical representation of answers to aid complex problem domain understanding. Autotrade was easily integrated into the ALCM framework during this contractual effort.

5.4.3 Framework Capabilities Summary

The ALCM framework prototype has met its objectives as presented in the Objectives (Section 3). The Framework Overview (Section 5.4.1) presented the near-term objectives in

the context of the framework implementation. The subsequent sections provided a road map of the capabilities within the prototype.

The following table (Table 20) provides a compact review of objectives satisfaction within the context of the prototype implementation.

Objectives	Prototype Features
	General capabilities
Risk analysis	Uncertain variables allowed on engineering estimates for automation implementation and potential
Temporal representation	Cost accounting and analysis options partitioned by life-cycle phase
Integrated documentation	Each model object has an attached documentation card that may include description, background, source, legacy, and explanation
Multiple fidelity levels	Models are amenable to decomposition to the lowest practical and required level of detail
	Output/display capabilities
Spreadsheet paradigm	Cost and WBS elements are related through the use of matrices
Graphics (histogram, line charts)	Model variables and resultant analyses are presented as a series of line and bar charts
Model structure	The model composition is graphically represented using hierarchical influence diagrams
	Comparative analysis
Manual vs. automation	Explicit accounting of the implementation method occurs with unique WBSs. Bottom-line analysis yields a single measure-of-merit for automation in a particular application

Table 20. The ALCM Framework Prototype Has Met the Program's Objectives.

6. Model Validation

6.1. Overview

Model validation for LCC models is a difficult problem. Projections for new programs are theoretically impossible to prove since the events have yet to occur. Complete model validation occurs only after the fact. Models executed in parallel with on-going programs are constantly being tested. However, getting to the point of being trusted again brings in the question of validation. How then, does one validate (i.e., gain a high level of confidence) an LCC model?

Our approach is to provide a plan, developed with NASA concurrence, that balances the desired, or required, level of confidence in the model solutions to the time and fiscal

constraints imposed. On the basis of the agreed foundation that models, by their very nature, abstract real world situations so that the solutions are tractable, ultimate validation will never be achieved. However, many steps may be taken that support a degree of confidence in the model's results. Our plan consists of a three level process listed below:

- **Construct Validation** — This level provides the lowest level of validation and a corresponding increase in the confidence of results. This level requires that the variables and algorithms contained in the model appear to be correct through desktop analysis. There is direct relationship among the degree of validation from this level to the expertise and prestige of the model reviewers. It is our intent to use recognized experts within Rockwell and NASA for this process.
- **Concurrent Validation** — This level provides higher degrees of validation than Construct Validation and has associated increases in result confidence. This portion of the process compares the results of the model in question to those models and data that are deemed to be equivalent. This is similar to testing to a "gold standard." The problems that this level generates are the identification of the comparable models/data, and the actual need for the model in question if other devices are available for use. It could be presupposed that the newer model has additional capabilities from the "trusted" model/data. This level is the goal for the demonstration product generated during this contractual effort.
- **Predictive Validation** — This level is highest that can be achieved in our process. An existing model generates, or the data already exists, several test cases for comparison to the model in test. The testing standard data is empirical; it is the result of actual program execution and associated costs. Autotrade uses this validation type. Several analogous programs were used as a basis for the calibration coefficients.

6.2. Sample Scenario

The tile re-waterproofing application provides the scenario to demonstrate ALCM framework prototype capabilities. Detailed modelling for the mobil base platform and manipulator system increases model confidence. The application provides the opportunity to provide proof-of-concept prototype examples. These capabilities include hierarchical model construction, use and comparison of estimation methodologies, temporal modelling aspects, and detailed model documentation.

Section 5.3 provides a narrative of the engineering system. section 5.4 contains several "screens" of the prototype while Appendix 9.3 includes rudimentary user notes.

7. Comments and Conclusions

The ALCM contract is progressing within the defined scope and funds established. Appendix 9.4 contains pertinent programmatic information. This report incorporates comments, questions, and suggestions from Technical Interchange Meetings #1 and #2. Appendices 9.1 and 9.2 contain copies of the briefing material for reference purposes.

An operations concept has been developed for the Automation Life-cycle Cost Model. This includes far- and near-term objectives. This report documents envisioned operational modes, user options, development phases, proposed products, and advanced technology plans. These topics were discussed during TIM #1.

An ALCM exists based on previous modelling efforts, known and obtainable data sources, and advanced technology impact determination. Development steps include architecture definition, reuse of existing models, DEMOS plan definition, and friendly user interface creation. This framework prototype was demonstrated during TIM #2.

An ALCM framework prototype exists. It capitalizes upon the robust DEMOS development environment including such features as hierarchical model decomposition, key parameter definition, historical data injection, and probabilistic model development. The implementation includes cost drivers and their relationships within the life-cycle. A friendly user interface provides access to probabilistic models, and table-driven and constant parameters. A robotic application scenario executes within the framework prototype.

DEMOS is an existing capability within Rockwell. Not only is the tool gaining widespread acceptance, training courses are springing up throughout the Corporation. The Rockwell Palo Alto Laboratory (RPAL) is promoting and assisting other Divisions in its use. RPAL is an active participant on this contract.

Table 21 contains the major comments and conclusions generated during this contractual effort.

- All phases of a system's lifetime must be examined to accurately determine life-cycle costs and potentials cost savings with the application of A&R.
- Examination of the all components within a space system supports Design to Life-cycle Cost (DTLCC) goals.
- A standardized approach to life-cycle cost models is possible with the use of accepted life-cycle phase definitions.
- Program-specific work breakdown structures provide flexible and powerful mechanisms for costing space systems potentially using A&R.
- Providing a variety of cost estimation methods improves model results and encourages analyst usage.
- A friendly user interface encourages iterative refinement of the life-cycle cost model providing improved results and increased confidence in those results.
- The ALCM framework is extensible with specialized modules as shown by the inclusion of the Autotrade software costing model.
- Decision analysis capabilities provide unique opportunities for review of life-cycle cost trade-offs.

Table 21. The ALCM Framework Examines the Entire System Life-cycle to Maximize Cost Savings Potential.

7.1 Suggested Improvements to the ALCM

First it should be stated that the technical objectives that we are pursuing, regarding the development of this cost model, are four-fold:

- (1) The model should be user friendly.
- (2) The model should support the design process and assist in making design decisions. That is, the user should be able to determine which hardware/software items are driving costs and should be able to perform trade studies and sensitivity analyses against a number of variables.
- (3) The model should be able to perform cost analysis according to the data available—via a top-down or a bottom-up approach.
- (4) The model should inherently possess as much related information as possible to: (a) minimize off-line calculations which introduce user/customer suspicions, (b) clearly document the system being modeled, and (c) depict the relationships between the variables and how the model is performing its calculations.

The following suggestions represent required/additional capabilities that could be added to the present ALCM model and will support the technical objectives outlined above.

- Provide system descriptors (i.e., mass and volume dimensions, fabrication material, etc.) at each hardware level of indenture.
- Put some, or all, of the descriptors mentioned in item (2) above, in the initial screens of the ALCM. Further, add such items as source of data, appropriate ranges of application, etc., to the menu of variable descriptors, in the ALCM.
- Provide a capability to generate Theoretical First Unit (TFU) costs for the various elements of the hardware breakdown structure within the model. This capability should be in two forms: (1) a top-down parametric approach, and (2) a bottom-up or accounting approach.
- Provide a relational table of common spacecraft materials and associated costing CERs for use with the generation of TFU and production costs.
- Provide "toggle switches," at the appropriate hardware indenture—similar to the switch between manual and automated options—for determining costs associated with top-down and/or bottom-up analysis approaches.
- Provide a capability at the appropriate hardware levels, for the automated option, to assess cost impacts of different levels of automation, technology and complexity factors.
- Integrate ALCM framework with database systems to maximize use of existing data.
- Provide a tool to facilitate cost engineers expressing expert judgment as probability distributions.
- Improve tools for automated report generation.
- Integrate with existing project management tools capable of Program Evaluation Review Technique (PERT), Gantt charts, and Critical Path Analysis (CPA).

- Improve tools for managing cost estimation libraries.
- Provide training materials for ALCM users including tutorials, on-line model explanation, guides for model extension, interpretation, and analysis.

Currently, the model treats the system and its inherent description independent of one another. A user cannot explicitly "see" alternative system descriptions and their impacts on costs, because the model currently deals only with "quantities" associated with a given system, not its descriptors. For instance, selecting titanium as an alternative material to aluminum is easily understood by a user, but the current model only depicts change numbers and the user does not necessarily know that titanium was selected over the aluminum. By including the system hardware description as a part of the model definition, a user will have clear visibility into what elements of a system are driving costs. Providing these system descriptors, as well as a characterization of the data fidelity and history, at the variable definition level (Figure 31) will allow a user to "browse" through the model to see exactly what the system is, how it is defined, and how it is being modeled.

The figure shows a software window titled "Object" with a standard Windows-style title bar. On the left side of the window is a vertical toolbar containing three icons: a balance scale, a grid, and a line graph. The main area of the window is a form with several labeled input fields, all represented by dashed-line boxes. The fields are organized as follows:

- Title:** A single-line text box.
- Description:** A multi-line text box.
- Definition:** A multi-line text box.
- Units:** A single-line text box.
- Inputs:** A multi-line text box.
- Outputs:** A multi-line text box.
- Material Type:** A single-line text box.
- System Weight:** A single-line text box followed by "(lbs)".
- System Volume:** A single-line text box followed by "(ft³)".
- Data Source:** A single-line text box.
- Data History:** A single-line text box.
- Application Range:** A single-line text box.
- Author:** A single-line text box.
- Last Save Date:** A single-line text box.

A large, bold, diagonal watermark reading "Example Only" is superimposed across the center of the form.

Figure 31. Sample ALCM Object Definition Window With Provisions For System Descriptions and Data Legacy/Fidelity.

In a parametric, or top-down analysis approach, several cost elements are factored from the TFU. Again, it will not be explicitly clear to the user what exactly is driving costs, because presently our model requires that the user calculate the TFU cost off-line. The model

construction enables the ALCM to calculate TFU costs. And depending on the sources of data and program maturity, the ALCM model should be capable of generating parametric or accounting-based TFU costs. These TFU costs need to be accomplished at the lowest appropriate levels of the hardware indenture, as depicted in Figure 32.

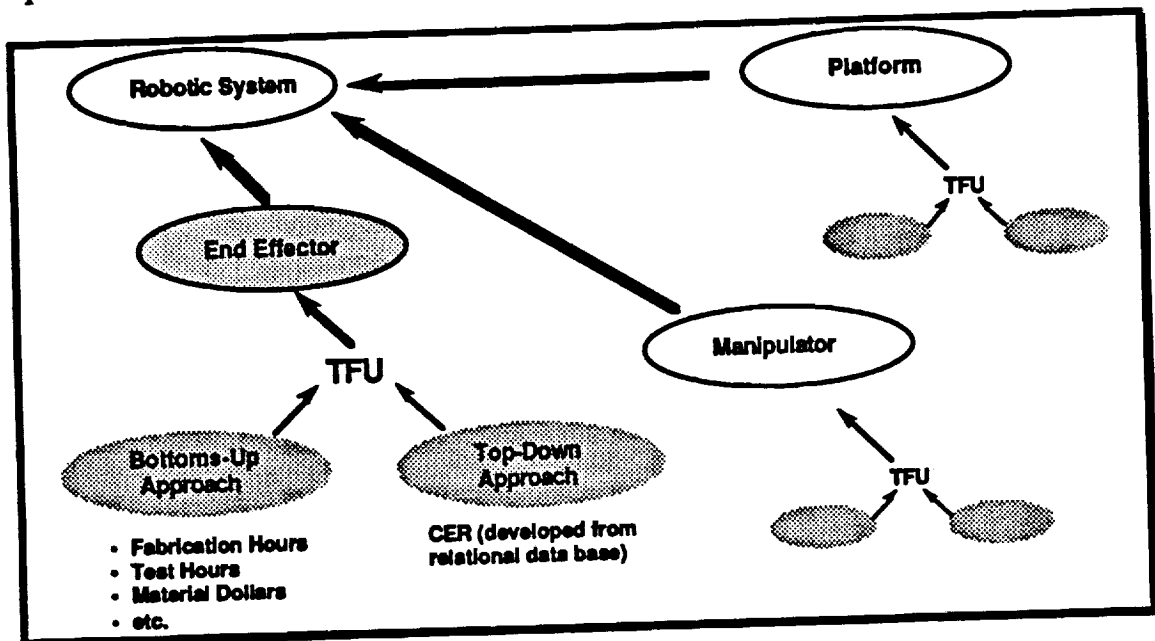


Figure 32. Parametric vs. Accounting Methods For Determining TFU Costs For The Various Elements of A Sample Robotic System.

It should be recognized that a truly "generic" life-cycle cost model is not practical, from the standpoint that the database containing all possible CERs would be enormous and nearly impossible to maintain. The GE PRICE H CER data base approaches this level, but is quickly becoming obsolete with the rapid advances in technology. It may be feasible to download the GE PRICE H data as an external module to the ALCM environment—however, in the next phase this would merely be a demonstration, and not a conclusion. The addition of a table of common materials (item #4) and CERs directly related to the "Material Type" field in Figure 31, is possible within the ALCM framework. This would not only add a "hint" of general application to the model, but would also make the model more user friendly.

Currently, if one is working with a top-down model, the required input must be of a parametric nature, and conversely, if the model is an accounting-based model, the input to support the model must be of an accounting nature. Unless a model is being generated for a specific purpose, a generic tool does not currently exist which can perform both analyses simultaneously. Adding the provision for looking at costs from either a top-down or bottom-up viewpoint (Figure 33) will give the Rockwell ALCM tool a capability not present in any existing model. This capability will enable a user to enter parametric data for discrete elements where only parametric data is available, and to enter accounting type data, for those discrete elements where there is sufficient historical accounting data. In addition, a

model possessing this kind of capability allows a user to assess, in real-time, the validity of his parametric estimates as he obtains the bottom-up data.

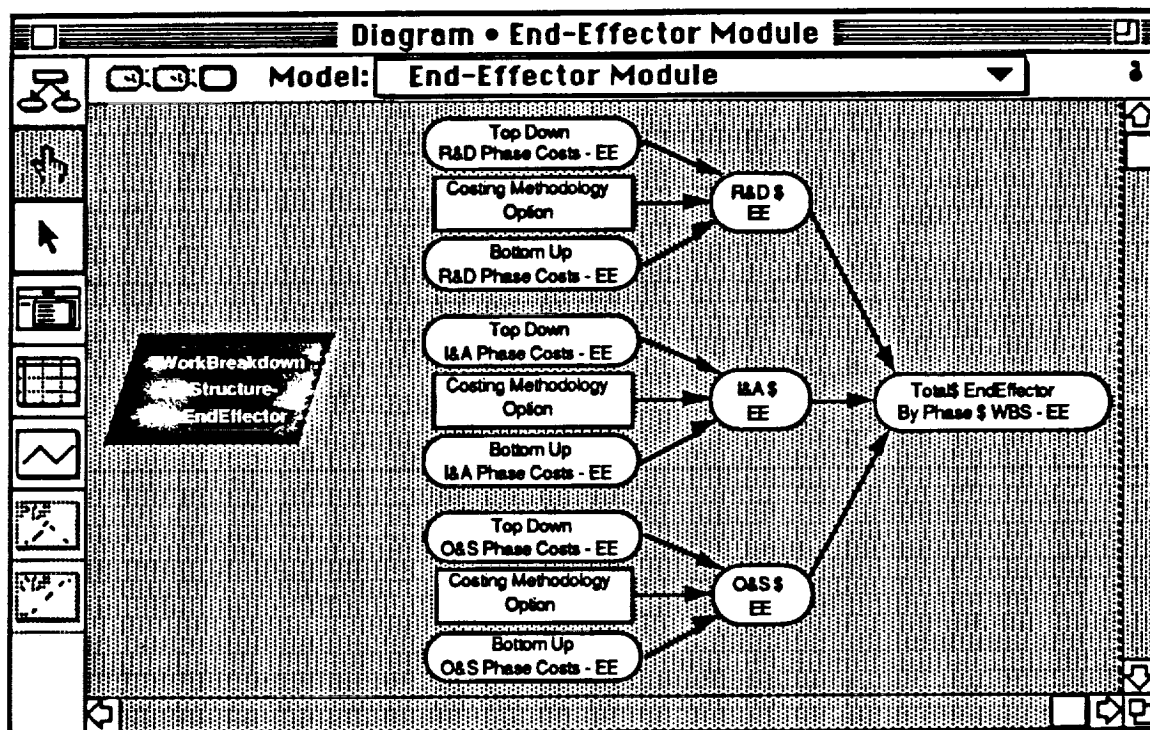


Figure 33. ALCM Model Will Perform Simultaneous Bottom-Up and Top-Down Cost Analysis and Assess Validity of Parametric Relationships.

NASA and DoD recognize different levels of automation (and robotics) and different levels of technology maturity. Presently, in the assessment of an automated mode, the model treats automation as only a single point scalar. This value can be changed to reflect varying levels of automation, but it must be calculated off-line. The model should possess a feature that would allow a user to assess various levels of automation (and technology maturity) for each hardware element of the automated option. This could be easily accomplished by adding a table/list of fixed options, and then calculate costs according to a determined algorithm relating level of automation, technology factors, level of complexity, etc. This capability is present, however, in the separable module Autotrade. The integration of this model within the ALCM framework has been investigated and can be accomplished with minimal risk. Autotrade is a proof-of-concept prototype for this feature.

7.2 Follow-on ALCM Effort Statement of Work

This effort's scope definition, implementation, and validation of a life-cycle cost model tailored to examining manual and automation method options for space systems. The follow-on effort to the Automation Life-cycle Cost Model (ALCM) encourages an iterative tool development. This involves the partitioning of effort into at least 3 separate development cycles. Each cycle contains requirement definition, design, prototyping, system

fielding, and analysis tasks. Events at the end of each cycle include a Technical Interchange Meeting (TIM), software demonstration, and associated documentation delivery. This iterative process provides ample opportunity for complete requirements understanding and mature product delivery. This contractual effort uses informal meetings and contractor-format documentation.

Task 1. Model Usage

This task identifies models for integration into the ALCM framework. Identified models may exist within and outside Rockwell International and NASA. Rockwell International provides model applicability judgment. Rockwell International accommodates suggested models from NASA within the scope of this contractual effort.

Subtask 1.1. Identify Model

This subtask identifies known models within and outside Rockwell International and NASA that support the objectives of the ALCM. A starting point is the Phase I contractual effort results. This subtask evaluates the applicability of identified models. Models need not be entirely applicable to the ALCM effort. Use of applicable model components is encouraged where practical.

Subtask 1.2. Evaluate Integration Potential

This subtask analyzes model integration potential. Areas factoring into the model potential include applicability to the ALCM, ease of integration, and framework utility.

Subtask 1.3. Integrate Models

Selection screening precedes integration of selected models into the ALCM framework. This integration results in software integration of separate modules or the redevelopment of the model within the ALCM framework environment. Software integration may include the expansion of ALCM and DEMOS capabilities.

Task 2. Data Usage

This task identifies data for Cost Estimating Relationships applicable for integration into the ALCM framework. Identified data may exist within or outside Rockwell International and NASA. Ease of data accessibility encourages maximum use of existing data within those two organizations.

Subtask 2.1. Identify Data

This subtask identifies known data within and outside Rockwell International and NASA that support the models in the ALCM framework. A starting point is the use of Phase I data and identified sources.

Subtask 2.2. Evaluate Database Integration Potential

This subtask investigates the integration potential of existing databases. The evaluation criteria may include computer platform, geographical location, amount of data, applicability to the ALCM objectives, and applicability to the existing framework.

Subtask 2.3. Create Access To Data

This subtask is responsible for allowing use of identified data within the ALCM framework. The data may be pre-existing or generated by a modelling capability. The data need not reside within the ALCM framework itself but, at a minimum, be accessible to it.

Subtask 2.3.1. Integrate Database Query Mechanism

This subtask involves the alteration of the ALCM framework (and possibly DEMOS) to accommodate the query mechanisms of the native database controlling the desired data. This integration effort results in a seamless environment for the user to access and manipulate pertinent data.

Subtask 2.3.2. Populate Internal Database

This subtask involves the use of database capabilities within the ALCM framework. Identified data populates this database. The ALCM framework provides data access mechanisms without reliance on external database modules. Manipulation of existing data to fit within existing ALCM (and DEMOS) constraints is within the scope of this task.

Task 3. Graphical User Interface Refinement

Graphical user interface (GUI) refinement accommodates changes in the operational environment originally envisioned for the ALCM. The Phase I effort identified new elements. Changes encountered during Phase II will be addressed and evaluated as they occur.

Subtask 3.1. Identify Graphical User Interface Requirements

The first step for refining the GUI is the identification of requirements. These may include new items generated during the Phase I effort and extensions required for the incorporation of models and data.

Subtask 3.1.1. New Requirements

New requirements may impose changes on the existing ALCM framework prototype. These requirements may have already been identified during the Phase I effort (e.g., system descriptors).

Subtask 3.1.2. Accommodate Extensions to Models and Data

The use of new models and data within the ALCM framework may create GUI requirements. Requirement consideration, on a case-by-case basis, includes the extent of changes to the existing system, system utility after the extension, and ease of change to the existing framework.

Subtask 3.2. Design Graphical User Interface Improvements

GUI design includes consideration for the identified requirements, computing platform and environment, and long-term objectives for the ALCM. The design capitalizes on the existing ALCM framework prototype capabilities. Minimization of major changes to the existing ALCM (and underlying DEMOS environment) reduce risk and contract cost. Full use of DEMOS capabilities provides a consistent and mature software system.

Subtask 3.3. Implement Graphical User Interface Improvements

The identified design changes result in the ALCM framework prototype being modified. Modifications to the ALCM framework and, possibly, the DEMOS environment provide new user capabilities and modes.

Subtask 3.4. Validate Graphical User Interface Improvements

Fielding the prototype system within Rockwell International and NASA organizations provides GUI validation for this contract. Identification and incorporation of appropriate end-user comments on system utility are products of this subtask.

Task 4. Model Refinement

The Phase II prototype incorporates areas of model refinement identified during the Phase I effort. These changes reflect analysis work accomplished during that phase. The changes require minimal alterations to the overall prototype architecture. The refinement extends the utility of the Phase I prototype within the scope of automation versus manual analysis.

Subtask 4.1. Expand Cost Analysis Capabilities

The Phase I framework prototype enforced a bottom-up cost analysis for the modelled system. This phase provides an expanded capability to select the analysis method based on the availability of data. Phase II prototype enhancements include support of bottom-up and top-down methods.

Subtask 4.1.1. Top-down Analysis

Access to cost estimating relationships applicable to the automated versus manual system supports top-down analysis. User discretion dictates CER usage. Selection of the appropriate CER and top-down method provide the Phase II prototype enough information to generate a life-cycle cost for the automated space system and manual alternative.

Subtask 4.1.2. Bottom-up Analysis

The Phase I framework prototype currently supports the bottom-up analysis method. The Phase II prototype retains this capability. User and customer comments may result in extensions to the current capabilities.

Subtask 4.2. Provide System Descriptor Capabilities

Phase II prototype improvements include definition and incorporation of system descriptors. This information includes, but is not limited to, material type, system weight and volume. This provides a summary level description of the system being modelled. Top-down analysis CERs use these data to generate component and system cost. These capabilities affect Subtask 3.3 (Implement Graphical User Interface Improvements).

Subtask 4.3. Provide Theoretical First Unit Generation Mechanism

Top-down analysis calculates several costs from the Theoretical First Unit (TFU) cost. The capability shall be provided to generate the TFU based on either parametric or accounting data. The Phase II prototype permits the association of a TFU to the lowest level hardware definition entered by the user.

Subtask 4.4. Create Common Materials Cost Estimating Relationship Table
Material type and weight are typically the basis of component CERs for space system costing. The availability of a table relating common materials used in space systems to accepted CERs enhances cost model performance. This is the start of providing cost analysis tools within the ALCM framework. This is a far-term objective of the ALCM. This subtask may use and integrate existing data. This capability affects Task 2 (Data Usage).

Subtask 4.5. Extend Autonomy/Automation Levels Capability Throughout Model

A far-term objective of the ALCM is to provide the ability for sensitivity analysis of the amount of automation provided within a space system. The Phase I prototype supports a discrete function for the hardware specification of a system. The software cost modelling partition permits the identification and analysis of several levels of automation. This subtask provides a consistent interface to this capability.

Subtask 4.6. Extend Software Cost Modelling Capability

The Phase I prototype contains a separable software costing module based on the COCOMO model. Phase II efforts fully integrate this module into the ALCM framework. System enhancements include additional modelling capabilities of COCOMO. The additions may include, but are not limited to, use of the detailed COCOMO model, implementation language sensitivities, and calibration features.

Task 5. Cost Estimating Relationship Creation

The scope of the Phase I effort permitted the limited generation of Cost Estimating Relationships (CERs) to support the execution of a demonstration costing scenario. Phase I also identified pertinent CERs as well as the need for additional relationships. This task expands on the Phase I effort to generate a library of CERs applicable to automated and manual space systems through the identification and creation of applicable equations.

Subtask 5.1. Generate Automation Cost Estimating Relationships for Specified Application(s)

This subtask creates CERs pertaining to automation as applied to space systems. Identification of terrestrial applications and their extrapolation to space and non-terrestrial domains is in the scope of this subtask. This subtask requires analysis of existing data within and outside Rockwell International and NASA.

Subtask 5.2. Generate Robotics Cost Estimating Relationships for Specified Application(s)

This subtask creates CERs pertaining to robotics as applied to space systems. Identification of terrestrial applications and their extrapolation to space and non-terrestrial domains is in the scope of this subtask. This subtask requires analysis of existing data within and outside Rockwell International and NASA.

Subtask 5.3. Generate Manual Cost Estimating Relationships for Specified Application(s)

This subtask creates CERs pertaining to manual methods as applied to space systems. Identification of terrestrial applications and their extrapolation to space and non-terrestrial domains is in the scope of this subtask. This subtask requires analysis of existing data within and outside Rockwell International and NASA.

Task 6. Scenario Management

A system description and associated set of cost data and relationships constitute a scenario. This task generates, executes, and analyzes scenarios within the ALCM framework environment. Rockwell International may generate or NASA may provide these scenarios.

Subtask 6.1. Scenario Generation

Scenario generation consists of the identification of a space system configuration and selection of various data and relationships. This subtask's accomplishments include documentation and unique identification of this data set. This subtask is responsible for the maintenance of the scenario set used in the ALCM.

Subtask 6.2. Scenario Execution

Transformation of a scenario into an executable form and the execution of the ALCM framework prototype on the data constitutes a scenario execution. This subtask requires documentation of a scenario and associated ALCM results.

Subtask 6.3. Scenario Analysis

Examination of scenario execution documentation and provision of results' synopsis comprise scenario analysis. This subtask provides interpretation of data validity and sensitivity. The analysis results are appended to, and maintained with, the previous scenario documentation.

Task 7. Report Generation

Report generation is at least once per development cycle in the form of working papers. These reports contain all pertinent information generated and documented by Rockwell International in support of this contract. The draft final report is in contractor format and contains the cumulative documentation from the interim reports to date. The final report, whose basis is the draft final report, incorporates questions, comments, and suggestions resulting from NASA review of the draft.

Task 8. ALCM Software

Rockwell International provides software comprising the ALCM to NASA that is capable of operating on a Macintosh II computer.

Task 9. Technical Interchange Meetings

Rockwell International supports NASA with the preparation and attendance of one Technical Interchange Meeting (TIM) per development cycle. At least one TIM will be conducted at Rockwell International's Space Systems Division (Seal Beach, CA facility). At least one TIM will be conducted at NASA ARC (Sunnyvale, CA).

8. References

1. Boehm, Barry W., *Software Engineering Economics*, Prentice-Hall, Inc., 1991.
2. Heer, Ewald and Henry Lum, *Machine Intelligence and Autonomy for Aerospace Systems*, Volume 115 Progress in Astronautics and Aeronautics, American Institute of Aeronautics and Astronautics, Inc., 1988.
3. Henrion, Max, John S. Breese, and Eric J. Horvitz, *Decision Analysis and Expert Systems*, AI Magazine, Vol. 12, No. 4, American Association for Artificial Intelligence, Winter 1991.
4. Henrion, Max, *DEMOS User's Manual*, Carnegie Mellon University, 1987.
5. Horton, William, *Illustrating Computer Documentation*, John Wiley & Sons, Inc., 1991.
6. NASA, *Space Station Freedom Automation and Robotics: An Assessment of the Potential for Increased Productivity*, March 1990.
7. Rockwell International IR&D Project 968, Autonomous Ground-based Control for Space-based Sensorcraft. *Ground Segment Autonomy Definitions, Allocations, and Assessment Working Papers*, 2/92.
8. United States General Accounting Office, *Military Space Operations: Operational Problems Continue With the Satellite Control Computer System*, GAO/IMTEC-89-56, (August 8, 1989).

9. Appendices

9.1. Technical Interchange Meeting (TIM) #1 Notes



100

100

100



100

100



AUTOMATION LIFE-CYCLE COST MODEL

TECHNICAL INTERCHANGE MEETING #1

DECEMBER 1991

AUTOMATION LIFE-CYCLE COST MODEL (ALCM)

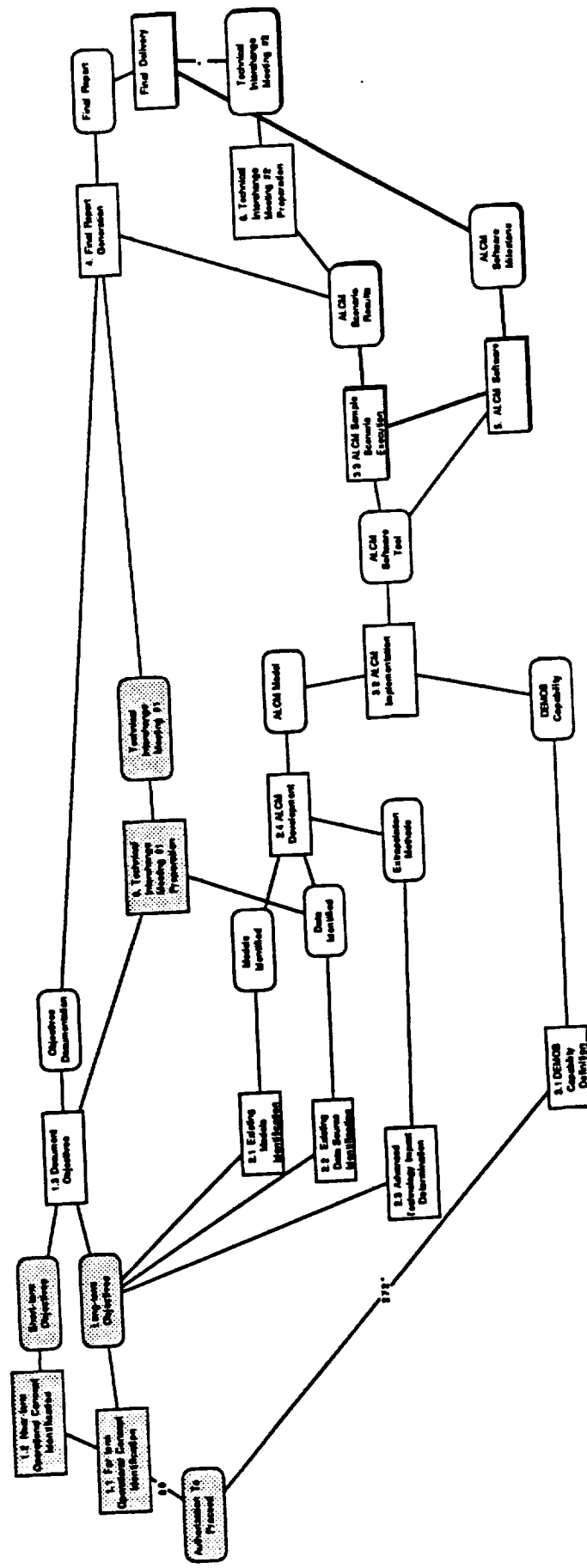


AGENDA

- INTRODUCTIONS
- PROGRAM OVERVIEW
- PROBLEM DEFINITION
- OBJECTIVES
- APPROACH
- ALCM TOOL
- MODEL VALIDATION
- ISSUES
- DEMONSTRATIONS

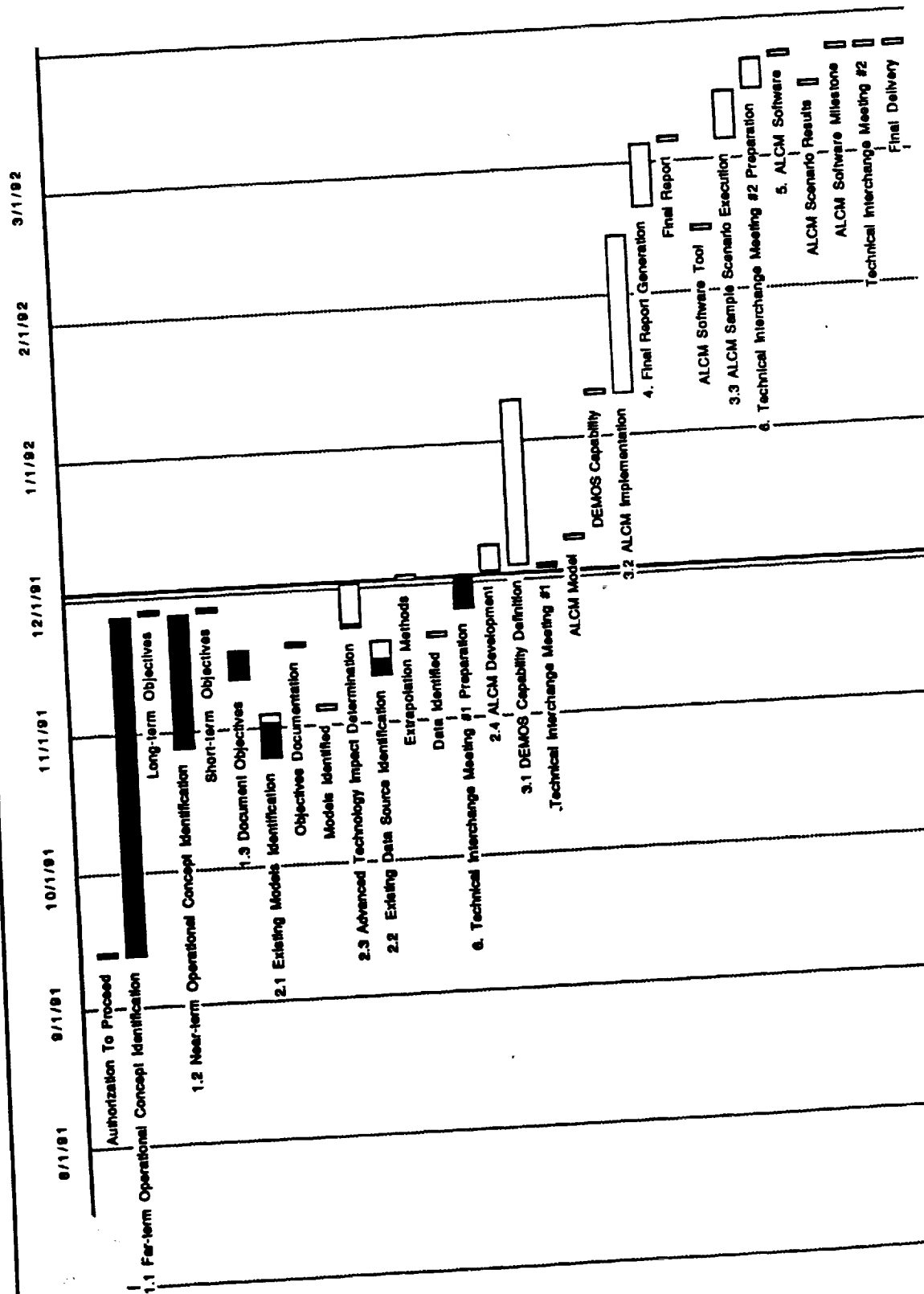


PROGRAM OVERVIEW — TASKS





PROGRAM OVERVIEW — SCHEDULE



PROBLEM DEFINITION

• COST ANALYSIS

- PARAMETRIC vs. EMPIRICAL
- CONFIDENCE
- VALIDATION

• COMPARATIVE ANALYSIS

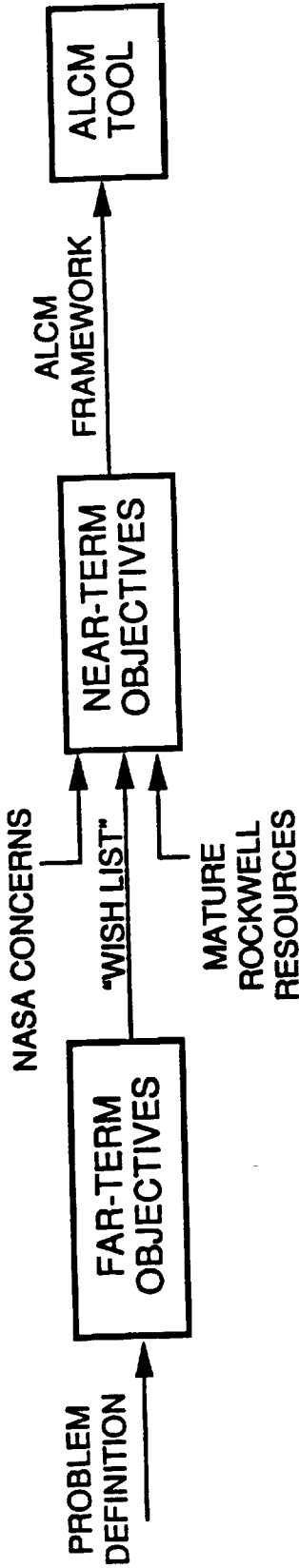
- MANUAL vs. AUTOMATION ALTERNATIVES
- AUTOMATION ALTERNATIVES
- PHASE/COMPONENT CONSIDERATIONS

• UNCERTAINTY ANALYSIS

- AVOID "POINT" DECISIONS
- IDENTIFY KEY SOURCES
- RELEVANT LEVEL OF DETAIL



OBJECTIVES



• GENERAL CAPABILITIES

- Assessment/Determination Tools
- Risk Analysis
- Driver Identification
- Temporal Representation
- Integration with Existing Databases

- Security of Data/Algorithms
- Integrated Documentation
- Multiple Fidelity Levels
- Uncertain Parameters

• OUTPUT/DISPLAY

- Spreadsheet Paradigm
- Graphical
 - histogram
 - pie
 - 2D/3D
- Model Structure
 - Line
 - uncertainty
 - values

• COMPARATIVE ANALYSIS

- Temporal (e.g., duration, frequency)
- Quantity
- Sparing
- Manual vs. Automation
- Level of Automation

■ : near-term

APPROACH OVERVIEW

- **ASSESS DESIRABLE CAPABILITIES**
 - STRUCTURE INTO NEAR- AND FAR-TERM
- **IDENTIFY USEFUL EXISTING MODELS/DATA**
- **IDENTIFY PARAMETERS FOR UTILITY ANALYSIS**
 - COST
 - SCHEDULE
 - ELEMENT
 - RELIABILITY
 - SUPPORTABILITY
 - OPERABILITY
 - INTEROPERABILITY
 - ...
- **DEFINE TOOL FRAMEWORK**
- **ENCODE SCENARIO THREAD FOR MODEL VALIDATION**
- **PROVIDE TRANSITION PATH FOR TOOL DEVELOPMENT**

APPROACH — COST/COMPARATIVE ANALYSIS

- **DOCUMENTED ENVISIONED TOOL CAPABILITIES**
 - GENERAL CAPABILITIES
 - OUTPUT/DISPLAY
 - COMPARATIVE ANALYSIS
- **INVESTIGATED EXISTING COST MODELS**
 - BRILLIANT EYES (BE)
 - OPTIMUM REPAIR LEVEL ANALYSIS (ORLA)
 - LCC1 (ROCKWELL B1-B)
 - PLANETARY LOGISTICS ANALYSIS AND EVALUATION TOOL (PLANET)
- **INVESTIGATED EXISTING SOURCES OF DATA**
 - ROCKWELL SHUTTLE DDT&E DATABASE
 - RSOC TASK MANPOWER TASKS DATABASE
 - ROCKWELL ROBOTICS DEPARTMENT
 - UNMANNED SPACECRAFT MODEL (V5 & V6) [AIR FORCE]
 - VARIOUS NASA PROGRAMS
- **CONCEPT DEVELOPED FOR ALCM TOOL STRUCTURE**
 - SYSTEM COMPONENT DEFINITION
 - LIFECYCLE PHASE
 - LEVEL OF AUTOMATION
 - LIBRARY OF TAILORED FUNCTIONS

APPROACH — MODEL/ALCM MATRIX

LEGEND

M/R Maintenance/Repair algorithms
A Approach
SA Spares Approach

Existing Models/Tools Potentially Applicable to ALCM

Brilliant Eyes Cost Model
(U.S. Air Force)
Optimum Repair Level Analysis
(ORLA - NASA)
Production Rate Adjustment
Factors (PRAF - U.S. Army)
Atlas (U.S. Air Force)
LCCT (Rockwell)
Planetary Logistics Analysis
and Evaluation Tool (PLANET)

Preliminary Definition of ALCM Modules

	Brilliant Eyes Cost Model (U.S. Air Force)	Optimum Repair Level Analysis (ORLA - NASA)	Production Rate Adjustment Factors (PRAF - U.S. Army)	Atlas (U.S. Air Force)	LCCT (Rockwell)	Planetary Logistics Analysis and Evaluation Tool (PLANET)
Design & Development	A					
Test & Evaluation	A					
Production	A					
Facilitization						
Launch Operations & Support		M/R,A				M/R
Orbit Operations & Support						
Mission Operations & Support		M/R,A		SA		M/R
Deorbit Operations & Support						
Landing Operations & Support		M/R,A				M/R
Disposal						



APPROACH — UNCERTAINTY ANALYSIS

DEMOS (Decision Modelling System)

- Influence diagrams display model structure
- Hierarchical structure helps organize large models
- Any uncertain value can be probabilistic
- Graphic parametric and probabilistic uncertainty analysis help identify key sources of uncertainty
- Non-procedural modelling language reduces programming effort
- Array abstraction allows flexible construction of multidimensional models
- Hypertext model with integrated documentation supports collaboration and sharing of models
- Libraries of functions support customizing for particular classes of application



APPROACH — DEMOS FEATURE USAGE

DEMOS FEATURE	ENVISIONED USAGE
HIERARCHICAL STRUCTURE	DEFINED MODEL STRUCTURE
PROBABILISTIC VARIABLES	UTILIZED FOR HIGHLY UNCERTAIN INPUTS
UNCERTAINTY ANALYSIS	IDENTIFY COST DRIVERS
MODELLING LANGUAGE	• PROVIDE FUNCTION FLEXIBILITY • COMPLEX MAPPINGS OF ELEMENTS IN LIFECYCLE
ARRAY ABSTRACTION	• MULTIDIMENSIONAL TRADE SPACE • INCORPORATION OF EXISTING DATA
INTEGRATED DOCUMENTATION	STRUCTURED PRESENTATION OF CRITICAL INFORMATION (VALIDATION, HERITAGE, EXPLANATION, ETC.)
LIBRARIES	DEVELOPMENT OF A TAILORED LIBRARY OF FUNCTIONS EXPLICITLY CONSTRUCTED FOR THE ALCM TOOL

C-2

ALCM TOOL

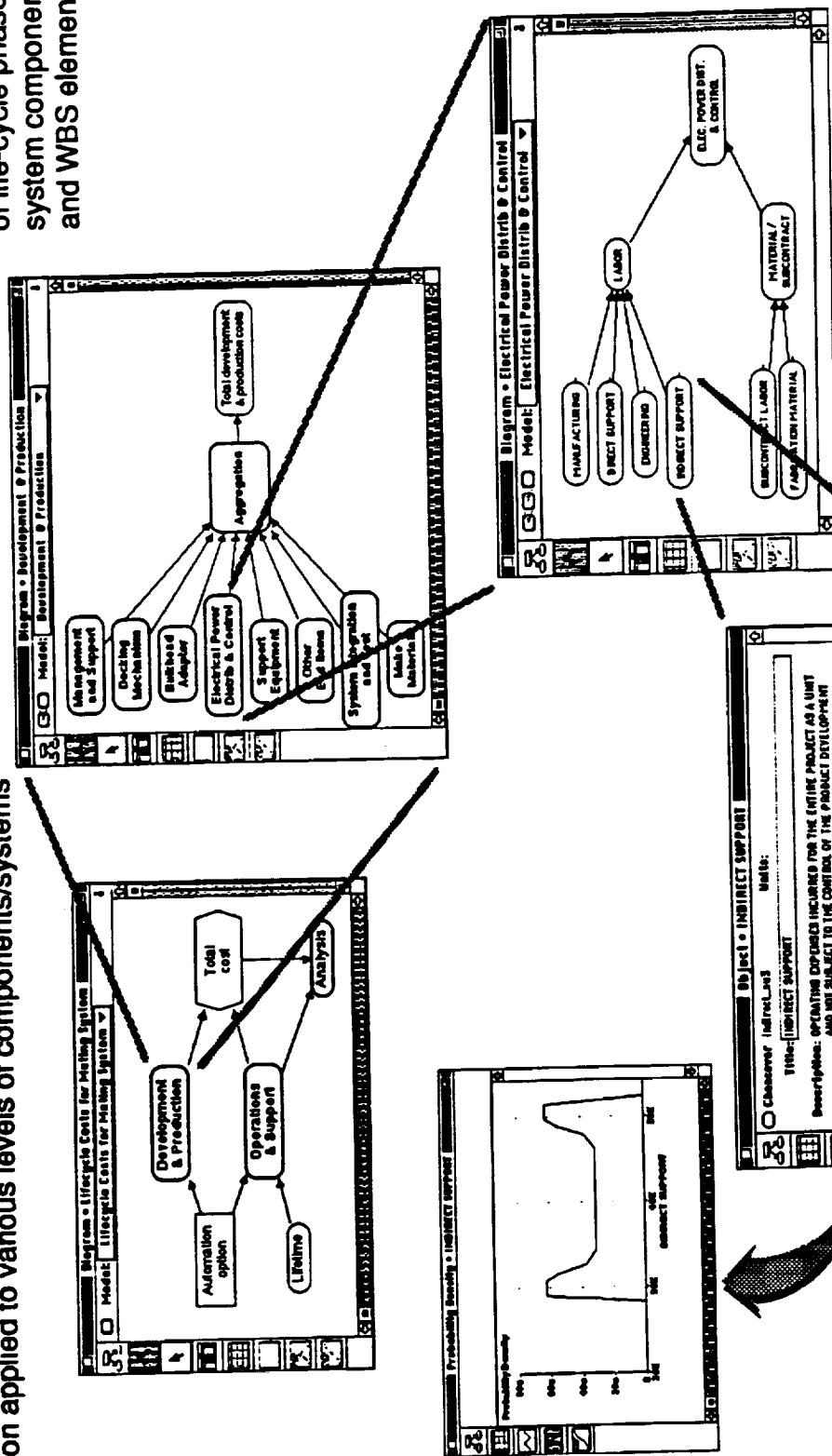
- DEVELOPMENT TO BEGIN IN CY1992
- HIGHLY SYNERGISTIC APPROACH TO BRIDGE 3 FUNCTIONAL ORGANIZATIONS AND 3 CAMPUSES
- MERGE CAPABILITIES OF DEMOS AND 4D MODEL
- PROVIDE FRAMEWORK FOR ENHANCED DEVELOPMENT
- PROVIDE EXECUTION "THREAD" TO PROJECT TOOL AND CONCEPT ATTRIBUTES
- CAPITALIZE UPON EXISTING BODY OF WORK

DIVERSE TALENTS WILL ENCOURAGE BEST TOOL TO MEET BROAD NEEDS

DEMO COMPARISON TO CONTRACT

- Levels of automation, as opposed to a discrete choice
- Automation applied to various levels of components/systems

Consistent presentation of life-cycle phase, system components, and WBS elements



Tailored library of functions for life-cycle cost domain

More detailed and consistent descriptive text for objects:

- description
- background
- source
- legacy
- explanation



MODEL VALIDATION

- PLANS TO ESTABLISH VALIDATION REQUIREMENTS
- 3 PHASE VALIDATION PROCESS
 - Φ1: CONSTRUCT VALIDATION
 - VARIABLES AND ALGORITHMS LOOK CORRECT
 - Φ2: CONCURRENT VALIDATION
 - COMPARE TO ACCEPTED MODELS/DATA
 - Φ3: PREDICTIVE VALIDATION
 - PREDICTED TEST CASES BASED ON EMPIRICAL DATA OR MODELS



ISSUES

- **MODEL APPLICATION SELECTION IS CRUCIAL WITHIN THE VERY NEAR-TERM (JAN. 1992)**
- **PREFER ACTIVE NASA PARTICIPATION IN MODEL DEVELOPMENT**
- **MODEL VALIDATION REQUIRES INDEPENDENT SOURCE OF DATA**
- **DELIVERABLE ITEM IS A FRAMEWORK; NOT A SPECIFICALLY TAILORED TOOL**
- **AVAILABILITY OF DATA ON THE MACINTOSH PLATFORM**
- **ROCKWELL ASSISTANCE FOR PROGRAM PROMOTION**



11



9.2. Technical Interchange Meeting (TIM) #2 Notes



Automation Lifecycle Cost Model (ALCM)

Final Technical Interchange Meeting

April 21 1992

**Tom Gathman
Max Henrion
and
Corinne Ruokangas**

**Rockwell International
Palo Alto Laboratory,
444 High St, Palo Alto, CA 94301**



Introduction to RPAL: The Rockwell Palo Alto Lab

- RPAL is a part of the Information Sciences Division of the Rockwell International Science Center, based in Thousand Oaks, California.
- RPAL specializes in research in advanced computer science, artificial intelligence, and decision analysis.
- RPAL was established in Palo Alto to foster close relations with Stanford, Berkeley, and other Bay Area centers of research excellence.



Current RPAL Projects

- Decision theory and AI planning for crisis planning (DARPA)
- VISTA: Intelligent user interface to display data for Space Shuttle flight controllers. Uses decision analysis to focus on what's probable and what's important (Rockwell Space Operations, NASA-JSC)
- Design Sheet: Reasoning with constraints for concept level design (DARPA)
- DEMOS: Development of applications for lifecycle costing, reliability analysis, technology assessment, concept level design tradeoffs (Rocketdyne, NAA, SSD, DCD, Collins)

Sample Demos applications within Rockwell

- Lifecycle cost analysis of upgrades to B1B.
- NASP system configuration cost and reliability analysis.
- Space Station shuttle berthing system.
- VLSI Chip manufacturing uncertainty analysis and design optimization.
- Lifecycle cost comparison of CO₂ removal technologies for long-period space flight.
- Intelligent Vehicle and Highway System (IVHS) technology cost-effectiveness analysis.



Agenda

- Introduction to ALCM Framework
- Background problems in cost modeling
- Basic issues addressed by ALCM framework
- Tile waterproofing robot: Sample application
- Demonstration
- Current status
- Possible follow-on ALCM development



Problems with conventional costing models

- (1) Assumptions, structure, and limitations are poorly documented.
- (2) The source, quality, uncertainty, and relevance of input data are unclear.
- (3) The limitations and uncertainties in the output results are not clearly communicated.
- (4) Models are *posthoc*, not developed as an integral part of the design process to help with early design decisions.
- (5) Separate cost models are built for different project phases and subsystems.
- (6) Tools tied to a single methodology, and so do not support cross-validation.
- (7) Poor documentation, incompatible assumptions and formats inhibit use of historical data to improve future cost forecasts

Primary goals of the ALCM Project

- **To develop a general framework for lifecycle cost modeling to evaluate automation and robotics for space systems**
- **To illustrate the framework by a specific application: The Shuttle Tile Waterproofing Robot.**
- **The focus of this contract is on the development and demonstration of the framework and general methodology.**



Issues demonstrated in the ALCM Framework

- Documenting and communicating model assumptions and structure
- Representing and analyzing uncertainties
- Providing analysis and results and multiple levels of aggregation, by subsystem, project phase, timing, and design alternative
- Combining multiple models
- Analyzing risks of schedule slippage

Combining multiple models

For alternative designs: E.g. Manual vs automated

Alternative costing methods: E.g. Top-down vs bottom-up.

Multiple models for different types of system: COCOMO for software vs. empirical CER for rocket engines.

Alternative levels of analysis: Simple aggregate model of primary systems vs. detailed models of subsystems.



Issues for a possible follow-on project

- Integration of ALCM modeling framework with database systems
- Tool to facilitate cost engineers expressing expert judgment as probability distributions.
- Improved tools for automated report generation
- Integration with existing tools (GANT chart, PERT, CPA) for project management.
- Improved tools for managing libraries for cost estimation
- Training material for ALCM users, including tutorials, on-line model explanation, guides for model extension, interpretation, and analysis.



AUTOMATION LIFE-CYCLE COST MODEL

TECHNICAL INTERCHANGE MEETING #2

APRIL 21, 1992

AUTOMATION LIFE-CYCLE COST MODEL (ALCM)

AGENDA

- PROBLEM
- OBJECTIVES
- APPROACH
- DELIVERED ALCM FRAMEWORK
- DEMOS USAGE
- Φ II STATEMENT OF WORK
- FINDINGS
- DEMONSTRATION

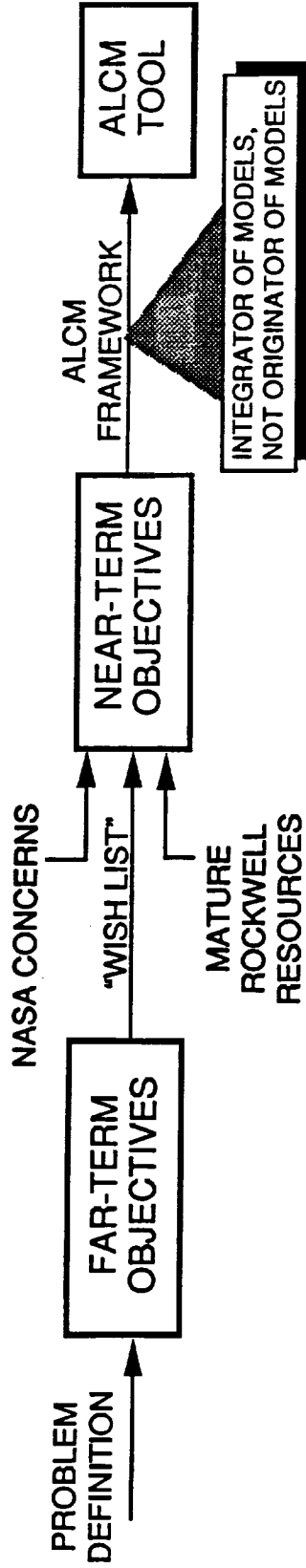


PROBLEM

- NO STANDARD COSTING METHODOLOGY APPLICABLE TO ENTIRE SYSTEM LIFE--CYCLE
- SPECIALIZED COST MODELS FOR SPECIFIC PORTIONS OF THE LIFE--CYCLE
- ASSUMPTIONS ARE "COMPILED" INTO LCC MODELS
- EASE OF MODEL MANIPULATION TO GAIN "CORRECT" ANSWERS



OBJECTIVES



• GENERAL CAPABILITIES

- Assessment/Determination Tools
- Risk Analysis
- Driver Identification
- Temporal Representation
- Integration with Existing Databases

- Security of Data/Algorithms
- Integrated Documentation
- Multiple Fidelity Levels
- Uncertain Parameters

• OUTPUT/DISPLAY

- Spreadsheet Paradigm
- Graphical
 - histogram
 - pie
 - 2D/3D
- Model Structure
 - Line
 - uncertainty
 - values

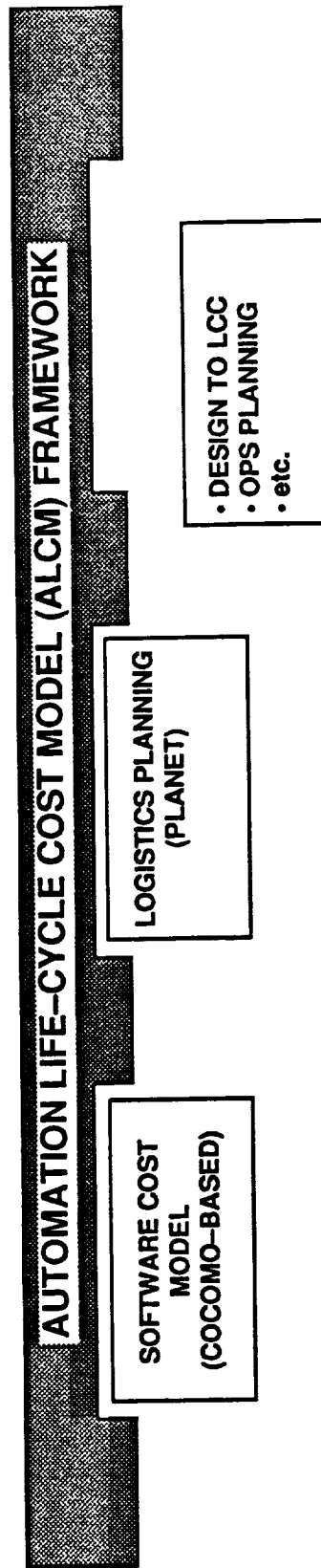
• COMPARATIVE ANALYSIS

- Temporal (e.g., duration, frequency)
- Quantity
- Sparing
- Manual vs. Automation
- Level of Automation

■ : near-term



APPROACH



FRAMEWORK CONSTRUCTION PERMITS:

- COST EVALUATION FROM THE SYSTEM LIFE-CYCLE PERSPECTIVE WHILE CAPITALIZING ON EXISTING, TRUSTED MODELS
- REVIEW OF COST BY MAJOR PHASE, COST CATEGORY, TOTAL SYSTEM LIFE-CYCLE
- MULTIPLE COSTING METHODS:
 - TOP-DOWN: CERS (LIBRARY)
 - BOTTOM-UP
 - ACCOUNTING
 - PARAMETRIC

ENCOURAGES COMPARATIVE
ANALYSIS OF METHODOLOGIES



DELIVERED ALCM FRAMEWORK

GENERIC FRAMEWORK COUPLED WITH SPECIFIC SHUTTLE ROBOTIC WATERPROOFING MODULES

- MAJOR PHASES
- COST CATEGORIES
- RATE DEFINITION
- COST ANALYSIS
- SCHEDULE SLIPPAGE
- TOP-DOWN vs. BOTTOM-UP
- SENSITIVITY ANALYSIS OF KEY PARAMETERS
- "FORM"-DRIVEN INPUT FOR COMPLEX MODELS
- INPUT PARAMETERS MAY BE DETERMINISTIC OR PROBABILISTIC

WATERPROOFING APPLICATION USED TO DISPLAY PROOF-OF-CONCEPT FRAMEWORK CAPABILITIES

- ACCOUNTING & PARAMETRIC BOTTOM-UP ESTIMATES BASED ON AVAILABLE DATA
- COCOMO-BASED SOFTWARE COSTING MODULE INCLUDING AUTONOMY AND REUSE PARAMETERS
- ADVANCED ROBOTICS FOR SPACE SYSTEM CERS INCLUDED IN TOP-DOWN METHODOLOGY



DEMOS USAGE

DEMOS FEATURE	USAGE
HIERARCHICAL STRUCTURE	DEFINED MODEL STRUCTURE
PROBABILISTIC VARIABLES	UTILIZED FOR HIGHLY UNCERTAIN INPUTS
UNCERTAINTY ANALYSIS	IDENTIFY COST DRIVERS
MODELLING LANGUAGE	• PROVIDE FUNCTION FLEXIBILITY • COMPLEX MAPPINGS OF ELEMENTS IN LIFECYCLE
ARRAY ABSTRACTION	• MULTIDIMENSIONAL TRADE SPACE • INCORPORATION OF EXISTING DATA
INTEGRATED DOCUMENTATION	STRUCTURED PRESENTATION OF CRITICAL INFORMATION (VALIDATION, HERITAGE, EXPLANATION, ETC.)
LIBRARIES	DEVELOPMENT OF A TAILORED LIBRARY OF FUNCTIONS EXPLICITLY CONSTRUCTED FOR THE ALCM TOOL

ΦII STATEMENT OF WORK

Task 1—Model Usage

- 1.1—Identify model
- 1.2—Evaluate integration potential
- 1.3—Integrate models

Task 2—Data Usage

- 2.1—Identify data
- 2.2—Evaluate database integration potential
- 2.3—Create access to data

Task 3—Graphical User Interface Refinement

- 3.1—Identify GUI requirements
- 3.2—Design GUI improvements
- 3.3—Implement GUI improvements
- 3.4—Validate GUI improvements

Task 4—Model Refinement

- 4.1—Expand cost analysis capabilities
- 4.2—Provide system descriptor capabilities
- 4.3—Provide theoretical first unit generation mechanism
- 4.4—Create common materials CER table
- 4.5—Extend autonomy/automation level capability
- 4.6—Extend software cost modelling capability

Task 5—CER Creation

- 5.1—Generate automation CERs for specified application(s)
- 5.2—Generate robotics CERs for specified application(s)
- 5.3—Generate manual CERs for specified application(s)

Task 6—Scenario Management

- 6.1—Scenario generation
- 6.2—Scenario execution
- 6.3—Scenario analysis

Task 7—Report Generation

Task 8—ALCM Software

Task 9—TIMs



FINDINGS

- STANDARD ANALYSIS DIMENSIONS ARE CONSTRUCTED THAT ARE APPLICABLE TO ENTIRE SYSTEM LIFE-CYCLE COST
- SPECIALIZED MODELS ARE APPLICABLE AND ARE, OR MAY BE MADE, COMPATIBLE WITH THE ALCM FRAMEWORK
- DEMOS ENVIRONMENT ENCOURAGES DOCUMENTATION OF MODEL PARAMETERS AND GRAPHICAL MEDIUM ENHANCES COMPREHENSION
- COMPARISON OF DIFFERENT METHODOLOGIES (e.g., TOP-DOWN vs. BOTTOM-UP, ACCOUNTING vs. PARAMETRIC) INCREASES CONFIDENCE IN RESULTS
- PROOF-OF-CONCEPT "PROTOTYPES" WITHIN THE ALCM FRAMEWORK ENCOURAGE FULL IMPLEMENTATION



9.3. ALCM Tool User Notes



Automation Life Cycle Cost Model for the Space Station and Space Shuttle: User's Manual

The key product of the NASA-Ames "Automation Life Cycle Model" contract with Rockwell Space Systems Division (SSD), is a Demos model, ALCM. This model was developed at Rockwell Palo Alto Lab (RPAL) and SSD to analyze life cycle cost and benefit of automation, as applied to a particular Space Shuttle ground-based task, Tile Waterproofing. The model demonstrates an integrated cost analysis tool, to support system design, project management, and risk analysis. In these days of scarce resources for aerospace projects, integrated treatment of these issues is increasingly seen as critical at NASA as well as DOD.

The ALCM model is implemented in Demos (Decision Modeling System) which is a general modeling environment for uncertain quantitative models. Demos runs on Macintosh computers, and was developed at Carnegie Mellon University. The ALCM model is illustrated by applying it to a specific task, waterproofing the Space Shuttle tiles between launches. The current technique for waterproofing tiles between shuttle launches is both risk and labor intensive. It requires injection of a highly toxic chemical into each tile by hand by a technician dressed in protective suiting while standing on a ladder. For safety, the operation can be carried on only during third shift. SSD is developing a robotic rover-based system to automatically locate each tile, and to apply the waterproofing substance. This operation can occur at anytime during the shuttle's ground based period, since the robot end-effector contains and quantifies the amount of the substance absorbed by each tile, and can meter the success of each application. While the initial R&D costs for the automated system exceed those for the manual system, the model suggests reduction in Operation & Support costs will result in significant payback in the lifetime of the automated system.

The ALCM was developed to compare the overall costs of the automated and the manual methods, over the major three phases of a project (R&D, I&A, O&S). The model includes a standard Work Breakdown Structure, the major phases with detailed cost elements for each phase, both top-down and bottom-up costing techniques, including Cost Estimating Relationships obtained from analysis of past projects, and the effect of schedule delays on overall costs. The model includes software cost analysis based on a DEMOS version of the COCOMO model, an industry standard.

Purpose of this Document

This document provides the user with the ability to navigate and explore the Tile WaterProofing model without requiring detailed knowledge of Demos. It is meant to be used in parallel with exe-



cution of the model on a Macintosh. It will support the user in understanding and demonstrating this model as a framework for Life Cycle Costing. It navigates through each of the major submodels of the main model, providing detail on specific nodes and their values. While the model runs on any Mac II, it is most effective for viewing on at least a 19" monitor; Demos will, however, relocate the windows appropriately for any size screen, including a PowerBook.

General Background for running the LifeCycleCost Model

There are several types of windows displayed within DEMOS, including the diagram, object, and value windows. The **diagram window** shows the overall relationships of objects, or nodes. These nodes include chance variables (rounded rectangles) which may be assigned probabilistic input values, decision variables (rectangles) over which the modeler has control, and submodels (dark outlined rounded rectangles) which allow the modeler to build his model in a hierarchical manner, with varying degrees of detail shown as appropriate. Arrows indicate the "influence" of one node on another. The top level diagram window is displayed when the model is opened; submodels may be displayed by double-clicking on the submodel node. For each object (i.e. node or diagram) there is an associated **object window**. The object window is accessed by double-clicking on a node, or by highlighting the node and selecting the object window icon from the left menu of the diagram window. The object window contains all attributes of the object, such as its class, variable name, title as it is displayed on the node in the diagram, description including any assumptions or qualifications, definition or relationship of this object to other objects which influence its value, and a list of any input and output variables. The user can request display of the variable's values as a table or graph format by clicking on view buttons in an object or diagram window, to produce what is known in this document as the **value window**. The value window displays the current value of the variable. Examples of all these windows are included in the following documentation.

It should be noted that the **attribute pane** of the diagram windows should be kept open while running this model. This can display an attribute, such as description, definition, or value of a selected node. It is opened by clicking on the mailbox flag in the lower left corner of a diagram window.

While representatives of all the submodels are expanded in this document, the greatest level of detail on how to navigate and evaluate the model is provided with the toplevel model and the first submodel, Tile WaterProofing Manual. Subsequent models will be expanded primarily with respect to their purpose in Life Cycle Costing, with fewer navigation details.

Top Level Diagram:

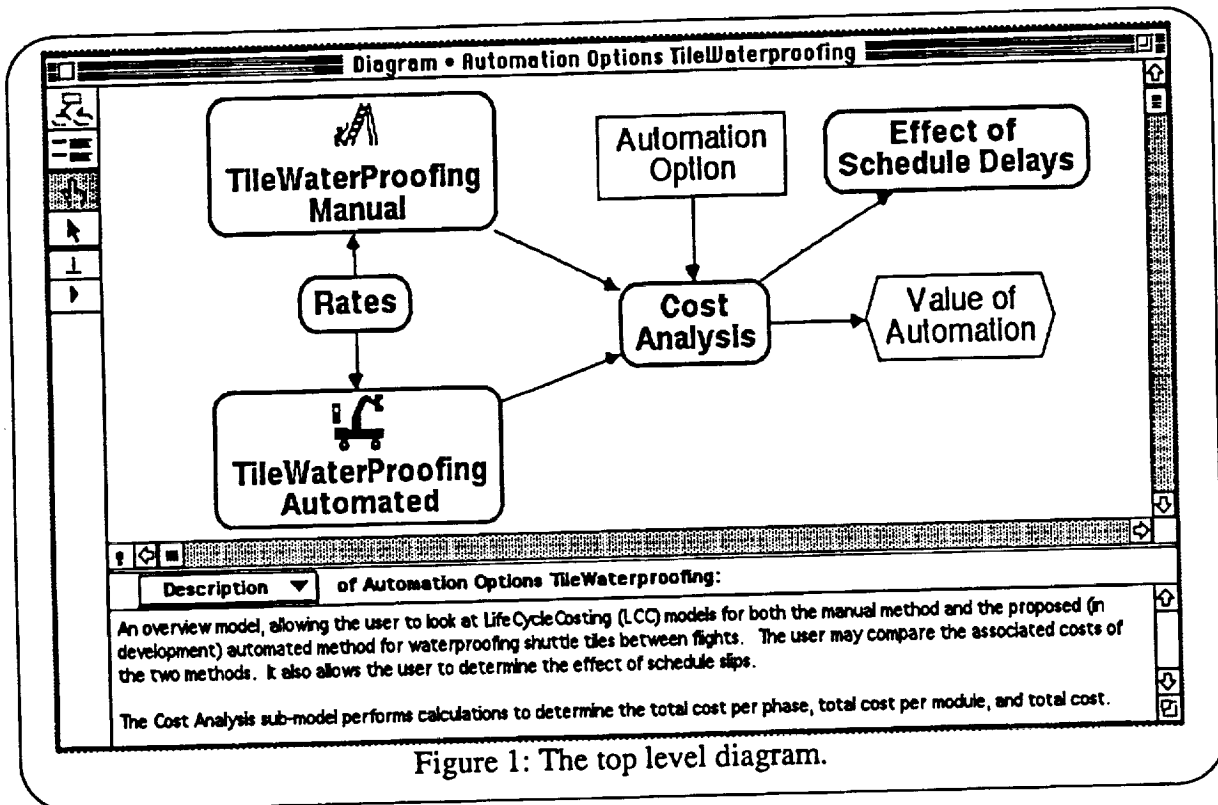


Figure 1: The top level diagram.

This diagram is displayed on initial startup of the model (initiated by double clicking on the file ALCM). It indicates the comparison of the Manual and the Automated techniques by performance of Cost Analysis over the life cycle of the projects. The **Manual**, **Automated**, and **Cost Analysis** nodes are all submodels, and can be displayed by double-clicking on them; they are expanded on subsequent pages of this document. In addition, the **Rate** submodel includes the information necessary to convert labor hours to dollar amounts, and the **Effect of Schedule Delays** expands to show the effect of timing delays on the overall costs of the project. The decision variable is **Automation Option**, i.e., whether to automate or not. The value variable, or objective, is the **Value of Automation**, which when displayed indicates the dollar amount that automation is expected to save over the manual technique. The object window for any node in the diagram window may be displayed by clicking on the object window icon on the left of the diagram window. The value of variables can be displayed by highlighting the variable's node, clicking on the right arrow from the icon menu to select the type of display, and clicking on the view icon to initiate the display.

The user should note the attribute pane displayed at the bottom of the diagram; since no specific node is currently highlighted, the description is that of the overall diagram. The contents of the pane may vary, by user choice. By clicking on the button **Description**, a set of other options is dis-



100

100

100

100

played, any of which may be selected. These options include all the attributes of the object currently selected, such as definition, name, class, etc. This attribute pane is very useful in navigating the model as it provides parallel descriptions of the elements of the model; it is opened by clicking once on the **mailbox flag** icon in the lower left of the diagram window.

Each of the submodels are expanded in subsequent pages. The highest level of detail on navigation, definition, and value display is incorporated in the first submodel, Tile WaterProofing, Manual Submodel. Subsequent submodels are described more with respect to their purpose and the relationships of their nodes.

Tile WaterProofing, Manual Submodel:

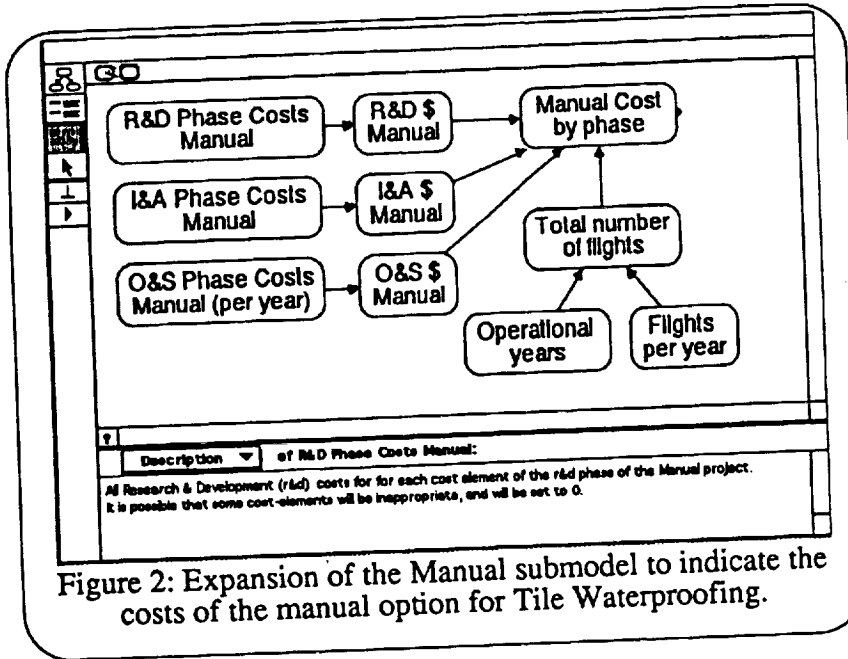


Figure 2: Expansion of the Manual submodel to indicate the costs of the manual option for Tile Waterproofing.

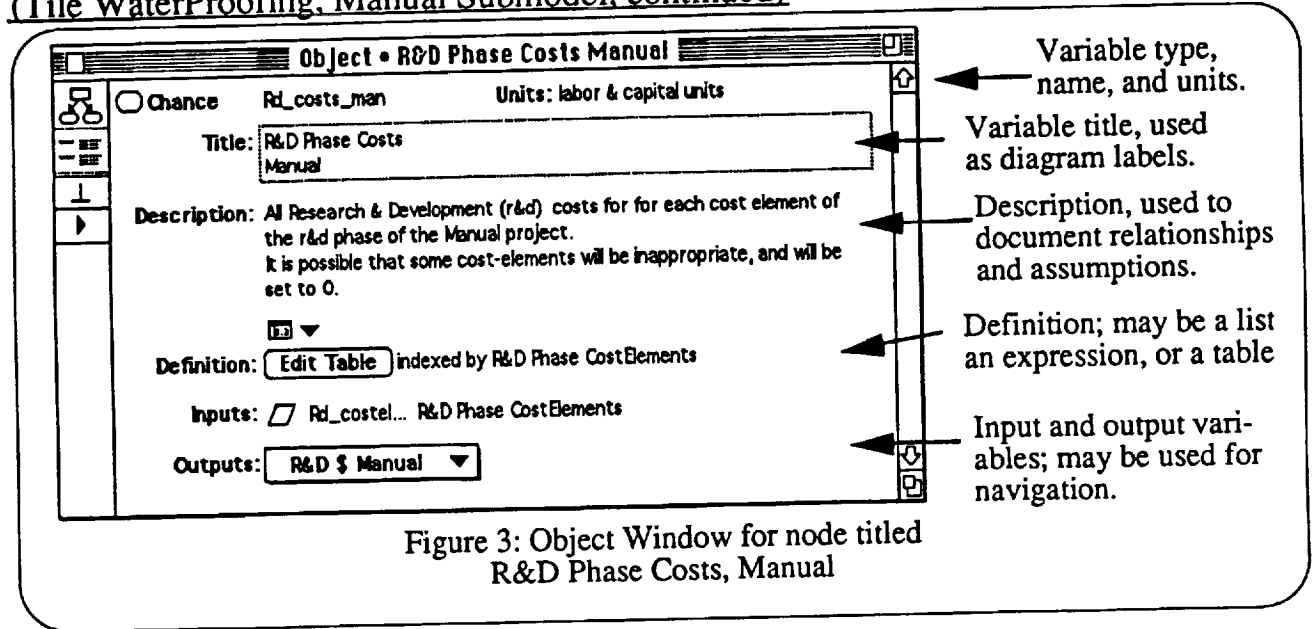
This diagram/submodel is displayed by double-clicking on the Tile WaterProofing Manual submodel of the top level diagram. It is at the second level of the hierarchy as shown by the hierarchy expansion icon at the top, second icon from the left. This indication of hierarchy is presented if the user has chosen **Show Model Hierarchy** in the preferences dialog of the edit menu.

This model provides entry of data in a bottom up method, where the modeler enters values for actual hours and capital costs for each of the three major phases of the manual implementation of the project. It should be noted that the detailed cost elements of each phase are identical for all projects, providing a consistent interface to the user. The modeler also enters the expected lifetime for the project by defining the values for **Operational years** and **Flights per year**. These values impact the O&S phase costs of the project.

Again, the attribute pane is open; in this case, since the **R&D Phase Costs, Manual** node is highlighted, a description of that node is displayed. By double-clicking on that node, or by selecting the object window icon when the node is highlighted, the object window for the node is displayed, as shown in figure 3 below.

As previously stated, this window documents all the attributes of the object, including its name, title, description, definition, and input and output variables. The actual values entered for this table may be viewed by selecting the "crossbar" icon, or may be modified by clicking on the Edit Table button of the definition. All the fields in the object window are either editable or are used as navigation aids. Input or Output variables, when selected, will display that variable's object window. The edit table is shown in figure 4.

(File Waterproofing, Manual Submodel, continued)



As previously stated, this object window documents all the attributes of the object, including its name, title, description, definition, and input and output variables. The actual values entered for the table definition may be viewed by selecting the "crossbar" icon, or may be modified by clicking on the Edit Table button of the definition. All the fields in the object window are either editable or are used as navigation aids. Input or Output variables, when selected, will display that variable's object window. The edit table is shown in figure 4.

Figure 4 shows the 'Edit Table' window for 'R&D Phase Costs Manual (labor & capital units)'. The table lists the following cost elements and values:

R&D Phase CostElements	
Development Eng (labor)	550
Producibility Eng&Plan(labor)	275
Tooling (capital)	0.46
Prototype Manufacturing (capital)	6.88
Data (labor)	800
System Test & Eval (labor)	6000
System/Proj Mgmt (labor)	10K
Training (labor)	40
Facilities (capital)	0
Software (labor)	0
Other (labor)	0

Figure 4: Input table for Manual R&D Costs, Bottom Up

This value window is accessed by clicking on the Edit Table button in the description of the node R&D Phase Costs, Manual; this button is found in the object window and in the attribute pane if Definition is selected.

For each cost element (Development Engineering etc) the modeler enters the appropriate value for either labor hours or capital dollars, in the bottom up approach to costing.



(Tile Waterproofing, Manual Submodel, continued)

As shown in the diagram window, figure 2, these values are then operated upon by the appropriate rate vector, and are summed with values from the other phases to determine the overall costs of the Manual project. These table entries may be entered as probabilistic or deterministic. To enter a probabilistic value, the modeler may type the distribution and its parameters directly into a table cell, or may select a template from the library menu, as shown for the definition of **Operational years** in figure 5 below. This use of a template from the library is to define a probabilistic function, but the modeler may select from a variety of existing function libraries to define variables.

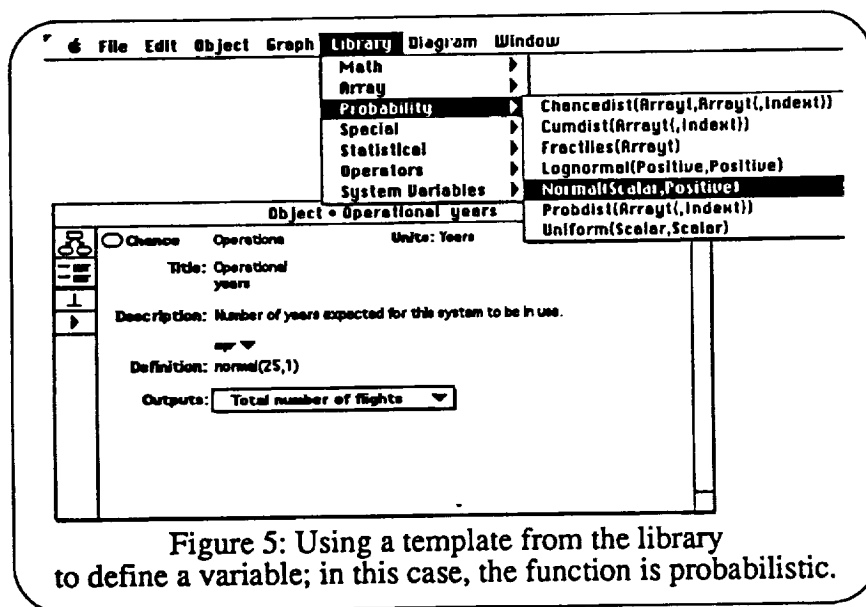


Figure 5: Using a template from the library to define a variable; in this case, the function is probabilistic.

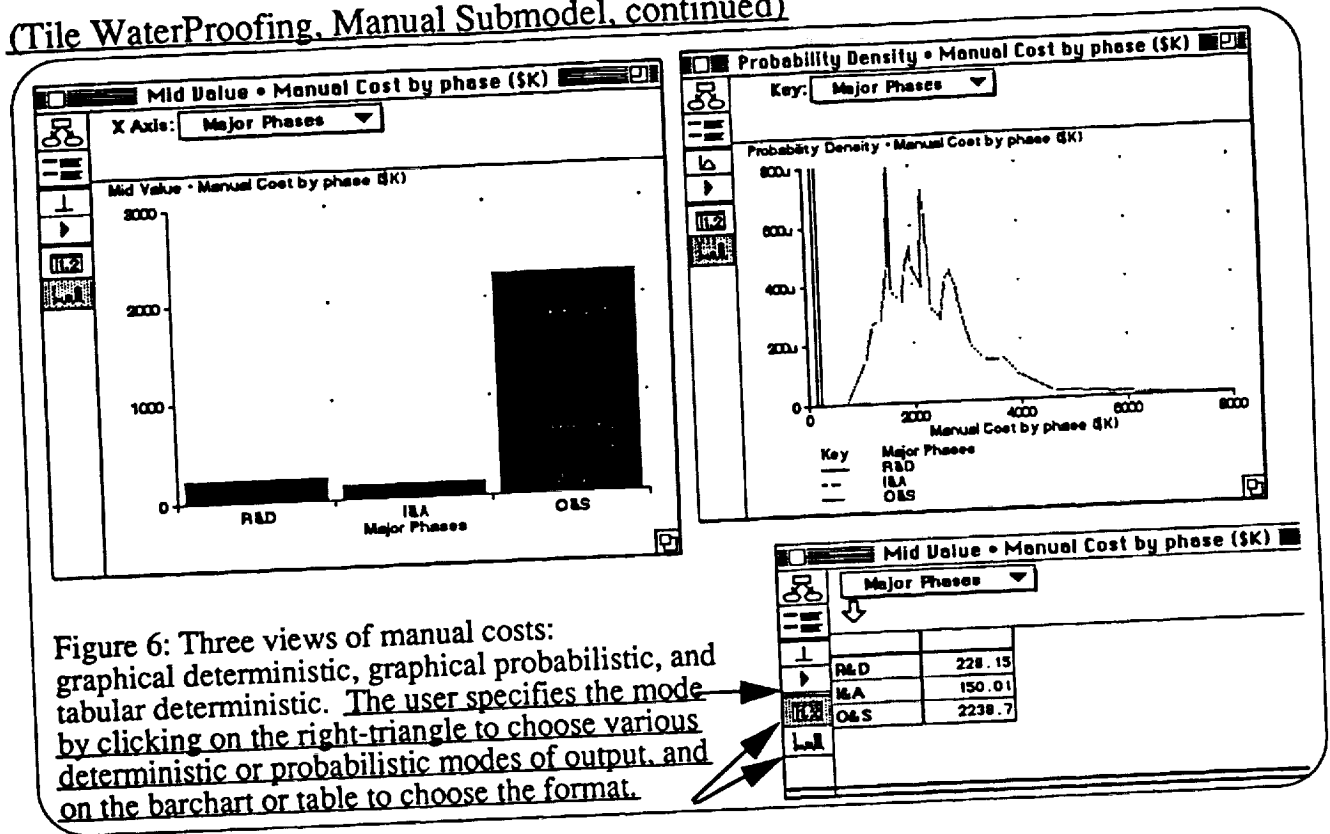
These function libraries include standard math, specific array functions, probabilistic, logical and other functions. The modeler selects a function template, which is inserted in his definition field; he must then correctly define the specific names of the input variables to the function.

At this point in this document, the user has been provided in-

formation on how to browse through diagrams and nodes, how to access the attributes of a node (object window or attribute pane), and how to enter values or definitions for variables. The user must also have the knowledge to access value windows within a diagram, and possibly to display the values in various modes. The value of the total manual cost of the project, organized by phase, is shown in figure 6 below.

The three windows of figure 6 indicate the value calculated for the Manual implementation of Tile Waterproofing, over the three major phases of the project. The first window is displayed when the user chooses the crossbar icon from either the diagram window (if the **Manual Cost by phase** node is selected) or clicks on the crossbar icon from the object window for the Manual Cost by phase node. This default window presents the data deterministically and graphically. [For a deterministic display, all probabilistic input variables are assigned their midvalue prior to performing the calculation of the specific variable's value; this provides a rapid calculation of the value, which may be refined to fully probabilistic if the user prefers.]

(Tile Waterproofing, Manual Submodel, continued)



The user may then choose a tabular display of the same data by clicking on the table icon of the graphical output. The values are then displayed in a tabular format, as shown in the bottom window of the figure. It should be noted that graphical value windows may be resized, and will scale their contents appropriately.

If the user prefers a full probabilistic analysis, he may specify that mode explicitly by clicking on the right triangle icon, and choosing from several options including Statistics, Probabilistic Density Function, or Cumulative Density Function. The default is MidValue, or deterministic. The top right window in figure 6 shows the probabilistic calculation for the three phases of Manual costs; in this case, both R&D and I&A are single valued, but O&S is a probabilistic variable, as indicated by the spread of its Probability Density Function (PDF).

This document has now summarized the techniques for navigating, defining, and displaying elements of models. The documentation of subsequent submodels will be limited primarily to describing the purpose of and the interactions within that submodel; a minimum of navigation information will be included. The submodels will be described in the order Rate, Cost Analysis, Effect of Schedule Delay, and finally TileWaterProofing, Automated, which is the submodel with the highest level of complexity.



100

100



Rate Submodel:

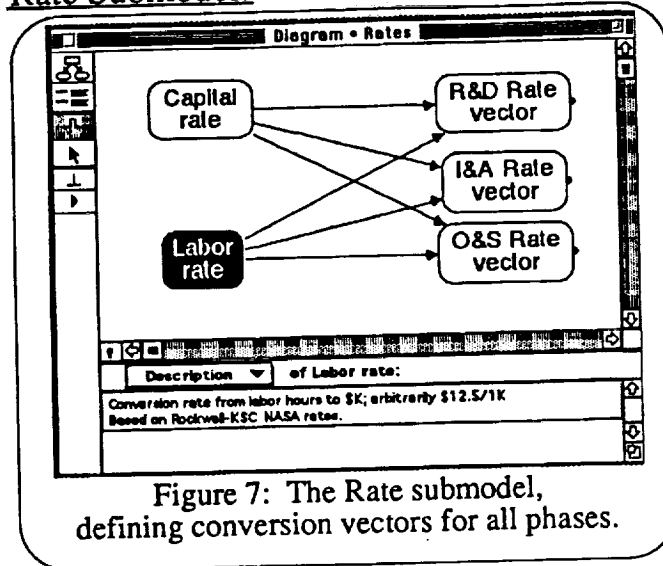


Figure 7: The Rate submodel, defining conversion vectors for all phases.

This submodel is accessed by double-clicking on the Rate node in the toplevel model. It provides a level of abstraction, where the modeler can separately define labor rates that will effect multiple segments of the overall project.

The variables defined in this model effect calculations in both the Manual and Automated submodel calculations, since they define the conversion rates from labor hours to Kdollars. For each cost element per phase, a conversion factor is defined. At this time, only two factors

are utilized; however, the number and level of granularity is dependent only on the modeler.

The cost elements vector for the R&D phase was shown in figure 4, above. In figure 8 below, the cost elements vectors for the other two phases are displayed. For O&S, the format is that of an entry table (accessed by clicking on the Edit Table button of the O&S Rate Vector definition), where the modeler has specified the use of Labor rate as the conversion factor for a small subset of the cost elements, with Capital rate as the primary conversion factor. Since capital values are entered directly in the Manual and Automated submodels as dollars amounts, no real conversion is required. However, the structure is maintained to indicate the flexibility of the Rate submodel. The value window is displayed for the I&A rate vector, to indicate the evaluated values of the assigned labor and capital rates within the vector.

The figure shows two side-by-side windows. The left window is titled "Edit Table - O&S Rate vector" and displays a table of cost elements with their respective conversion rates. The right window is titled "Mid Value - I&A Rate vector" and displays a table of cost elements with their evaluated values.

O&S Phase CostElements	Choose Indices
Personnel (labor)	Labor_rate
Mission O&S (labor)	Labor_rate
Orbit O&S (labor)	Labor_rate
Descent O&S (labor)	Labor_rate
Landing O&S (labor)	Labor_rate
Maintenance-Labor (labor)	Labor_rate
Maintenance-Material (capital)	Capital_rate
Maintenance-Transportation (capital)	Capital_rate
Spares (capital)	Capital_rate
Training (capital)	Capital_rate
Documentation (capital)	Capital_rate
Support & Test Equip (capital)	Capital_rate
Software Maint (labor)	Labor_rate
Disposal (capital)	Capital_rate
Other (labor)	Labor_rate

I&A Phase CostElements	
Non-recurring (capital)	1
Production (capital)	1
Eng Changes (capital)	1
Syst Y&E (capital)	1
Data (capital)	1
System/Project Mgmt (labor)	0.0125
Oper/Site Activation (capital)	1
Training (capital)	1
Initial Spares & Repair (capital)	1
Transportation (capital)	1
Facilities (capital)	1
Other (capital)	1

Figure 8: Rate vectors as entered (for O&S cost elements) and as evaluated (for I&A cost elements).



Cost Analysis Submodel:

This model is accessed by double-clicking on the Cost Analysis node of the toplevel model. Its purpose is to combine the costs determined in the Manual and Automated submodels, and to allow

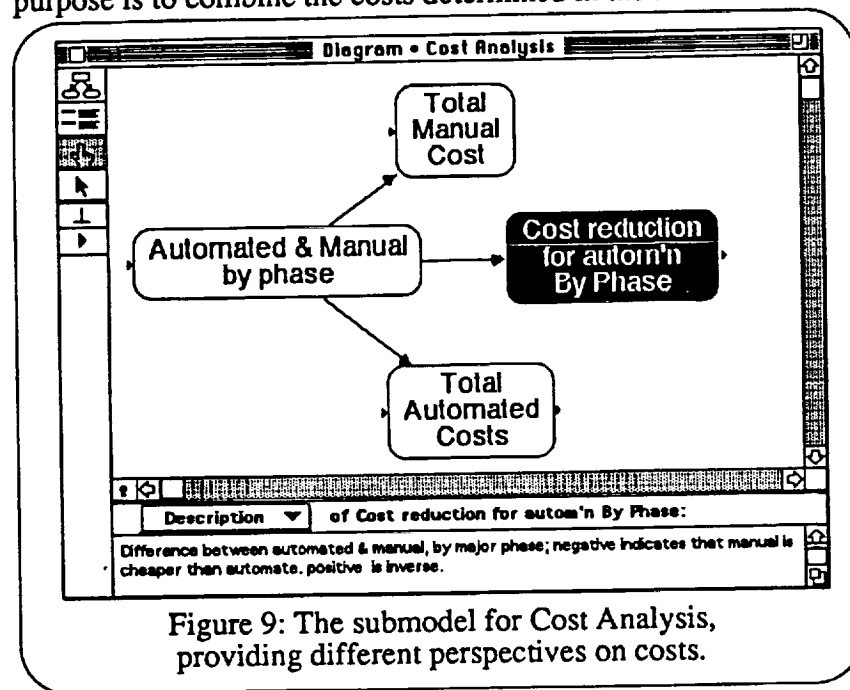


Figure 9: The submodel for Cost Analysis, providing different perspectives on costs.

the user to view these values from multiple perspectives. The difference value is used as output to define the overall value variable of the toplevel model. Total Automated Costs are used as the basis of calculations in the Effect of Schedule Delays submodel.

As indicated in figure 9, costs are derived from those overall costs determined per phase in the Manual and Automated submodels, and may be displayed in

conjunction, or by technique (Manual or Automated), or as the difference between the two techniques.

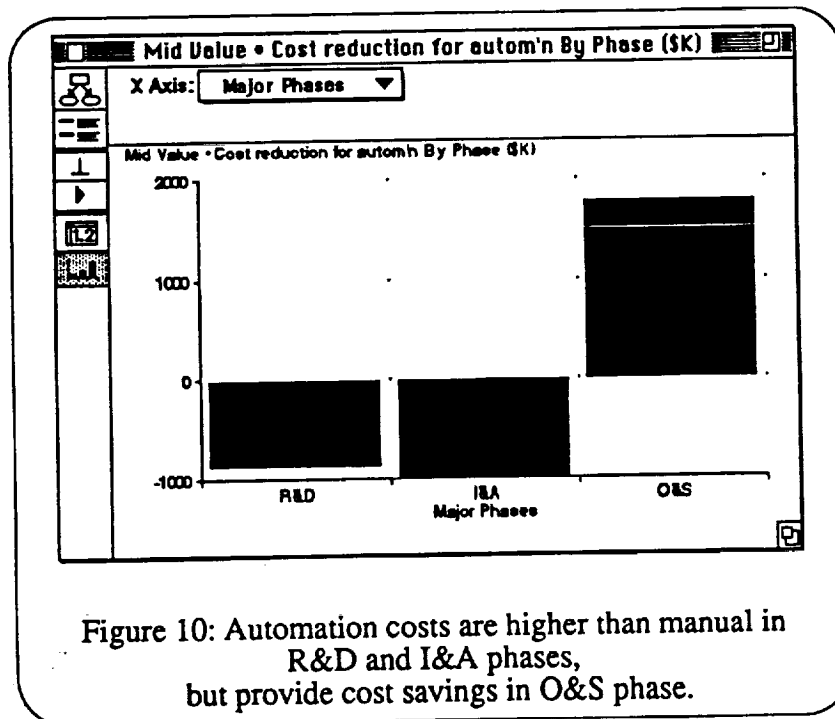


Figure 10: Automation costs are higher than manual in R&D and I&A phases, but provide cost savings in O&S phase.

Figure 10 displays the final value calculated in this submodel, the difference between Manual and Automated costs for each phase.

While it is displayed in graphical deterministic format, it could be displayed in tabular form. If there are any probabilistic variables, it may be displayed in probabilistic mode as well.



Effect Of Schedule Delays Submodel:

This submodel, accessed by double-clicking on the Effect Of Schedule Delay node of the toplevel diagram, determines the effect of a delay applied to the overall assigned schedule of tasks, explicitly with respect to the Automation project. While it could be expanded to include assignments of different delays to various parts of the project, this relatively naive treatment provides proof of the concept of dealing with uncertain schedules.

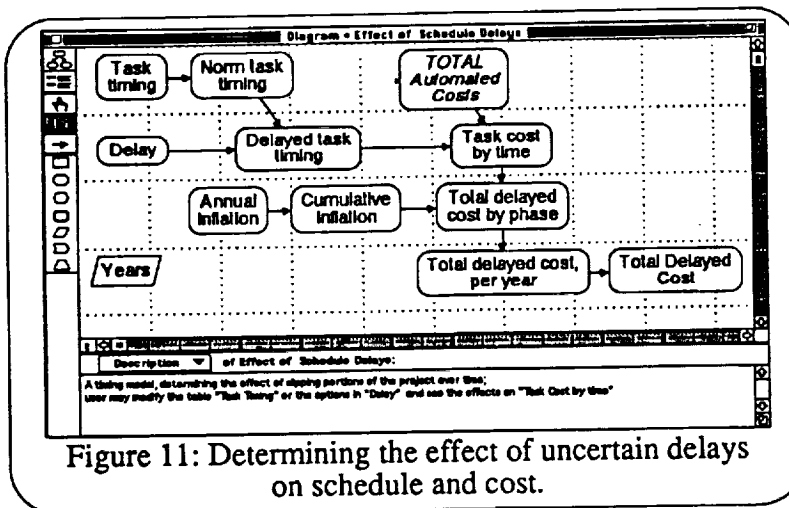


Figure 11: Determining the effect of uncertain delays on schedule and cost.

As indicated in the attribute pane of figure 11, this is a timing model, for determining the effect of slipping the project over time for uncertain amounts of time; the user may modify the initial timing schedule contained in the input node Task Timing or he may change the delay values in Delay and then may observe the effects on Task Cost by time which simply shifts the expected

costs over the available years. Further, he may view the effect on costs of inflation combined with the possibly uncertain delays. Figure 11 displays the diagram window; figures 12-14 display the values as effected by the delays.

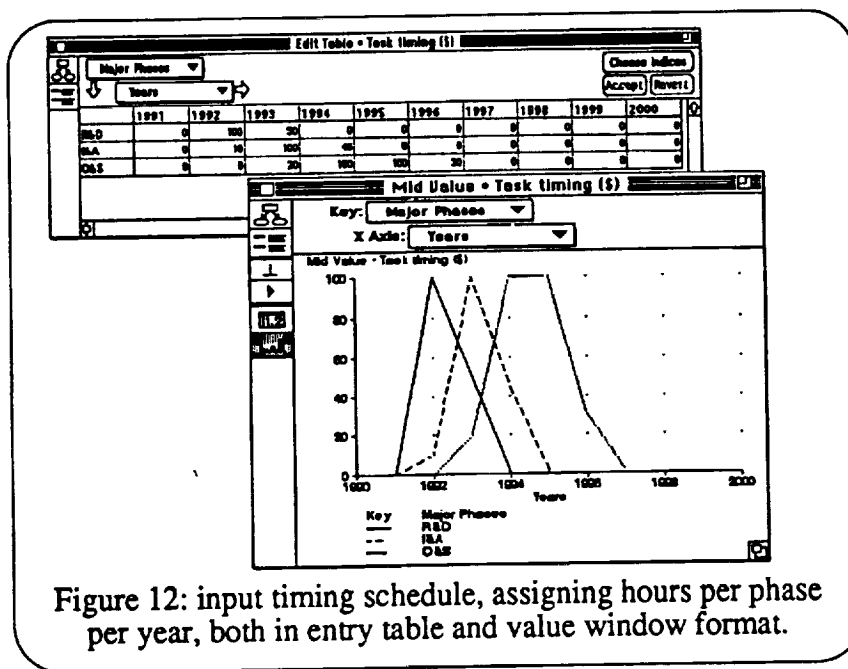


Figure 12: input timing schedule, assigning hours per phase per year, both in entry table and value window format.

Figure 12 indicates the schedule as initially entered by the modeler, both in tabular and graphical format. The modeler enters the hours expected per year per phase. These values are then normalized, and the delay is applied to them, yielding the Task cost by time delays. In this case, two values were entered for Delay by the modeler: 0, i.e., no change, and Lognormal (1,1.5) indicating a probabilistic delay ranging between approxi-



(Effect Of Schedule Delays Submodel, continued)

mately .5 year and 3 years. Task cost by time is then identical to the initial task timing, in the first case, and is shifted right in the second case. The effect of inflation is then taken into account, yielding the value windows shown in figure 13.

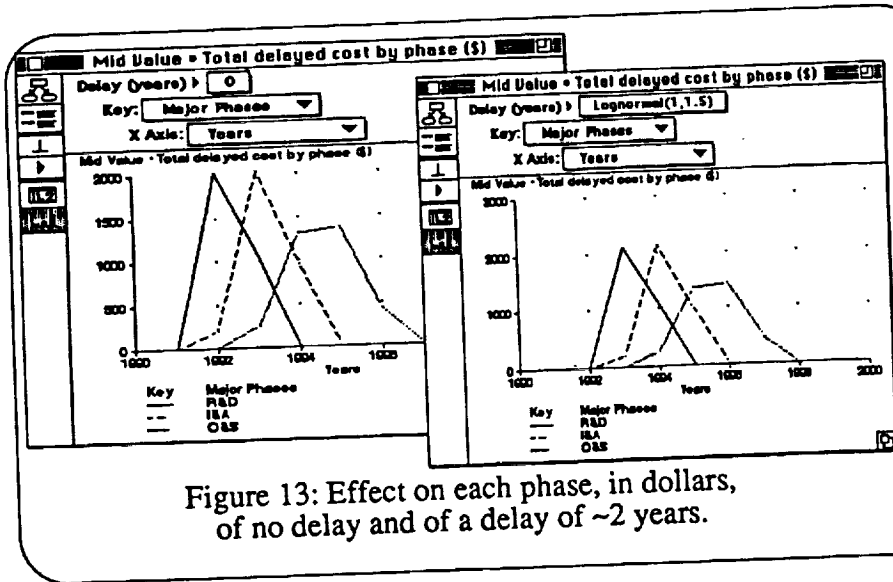


Figure 13: Effect on each phase, in dollars, of no delay and of a delay of ~2 years.

As indicated, the dollar values for a non-delayed project peak near \$2000K for both R&D and I&A, while O&S peaks near \$1500K. However, application of the second delay yields peak values closer to \$2200K. These displays can also be presented in probabilistic or tabular format.

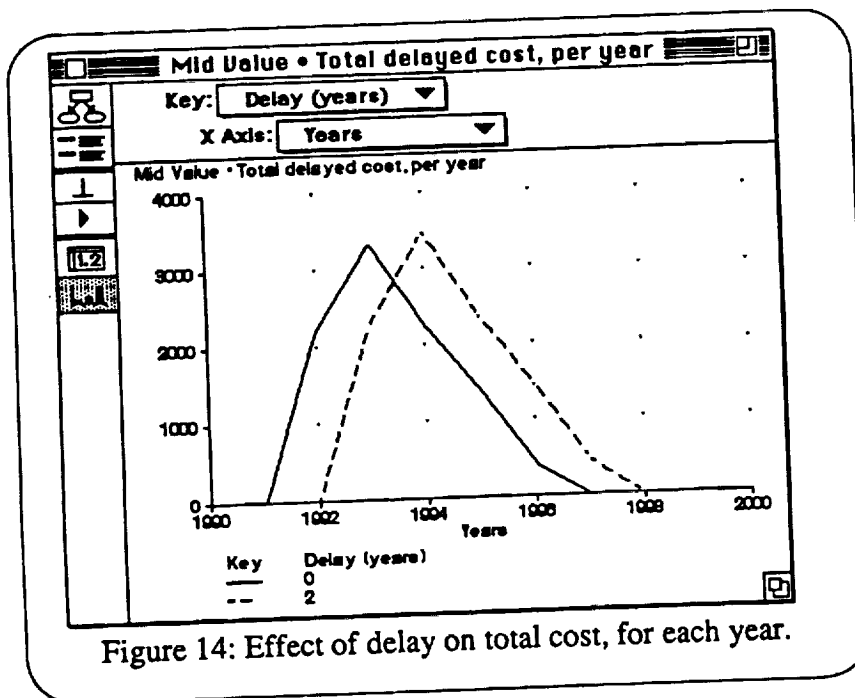


Figure 14: Effect of delay on total cost, for each year.

The effect of the delay on total cost is shown in figure 14, for each year.

As indicated, the values are summed over all phases, and are delayed in time as well as incremented in total cost value.

Although this submodel presents Delay as a single value effecting all phases, it could readily be expanded to specify independent delay values for each phase, or elements of each phase.



100-100-100



100-100-100

100-100-100

100-100-100



Tile WaterProofing, Automated Submodel:

This submodel provides the most complexity of all the submodels in the toplevel model. It contains submodels that utilize bottom-up techniques based on "Theoretical First Unit" values determined from previous projects (see Bottom-Up Cost Calculation submodels EndEffector, Manipulator and MobileBase) as well as top-down techniques using cost estimating relationships (CERs) based on analysis of previous projects (see Top-Down Cost Calculation). In addition, within the Bottom-Up submodel, the COCOMO model is incorporated for determining the cost of software associated with the WorkcellController. This model, an industry standard for calculating software costs, has been converted to Demos format.

This variety of techniques for determining the expected costs of the major phases of a project demonstrates this LifeCycleCosting model as a framework for assessing and verifying overall project costs. While these techniques are varied, the framework could be enhanced even further by the incorporation of other costing techniques. It is expected that other techniques will be incorporated during a follow on effort.

This submodel is accessed by double-clicking on the **Tile WaterProofing, Automated** node of the toplevel model. It indicates that both bottom-up and top-down costing techniques will be used to

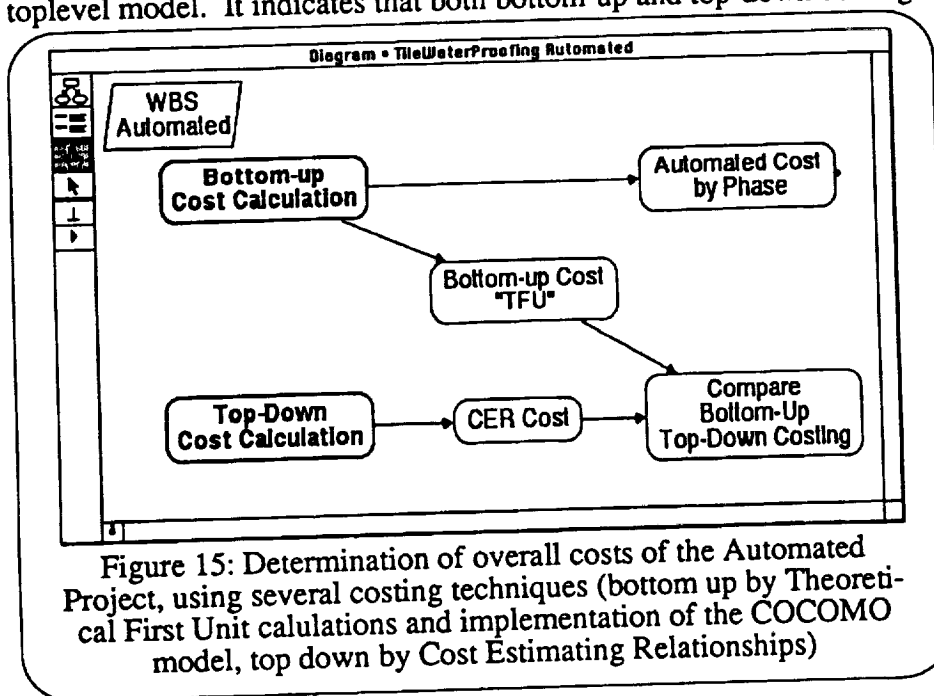


Figure 15: Determination of overall costs of the Automated Project, using several costing techniques (bottom up by Theoretical First Unit calculations and implementation of the COCOMO model, top down by Cost Estimating Relationships)

determine overall costs of the Automated technique. The theoretical first unit (TFU) based calculations are then compared with the CER calculations. In this case, the **Automated Cost by Phase**, as based on bottom-up calculations, is used by the Cost Analysis submodel of the toplevel model.

Details of both submodels, as well as the result-

ing values are summarized in following sections. For the Bottom-up submodel, only one example of the TFU calculations will be shown; it is left to the user to browse the other similar submodels. The final costing for the automated technique is included at the end of this document.

Bottom-Up Cost Calculation Submodel of Automated Submodel:

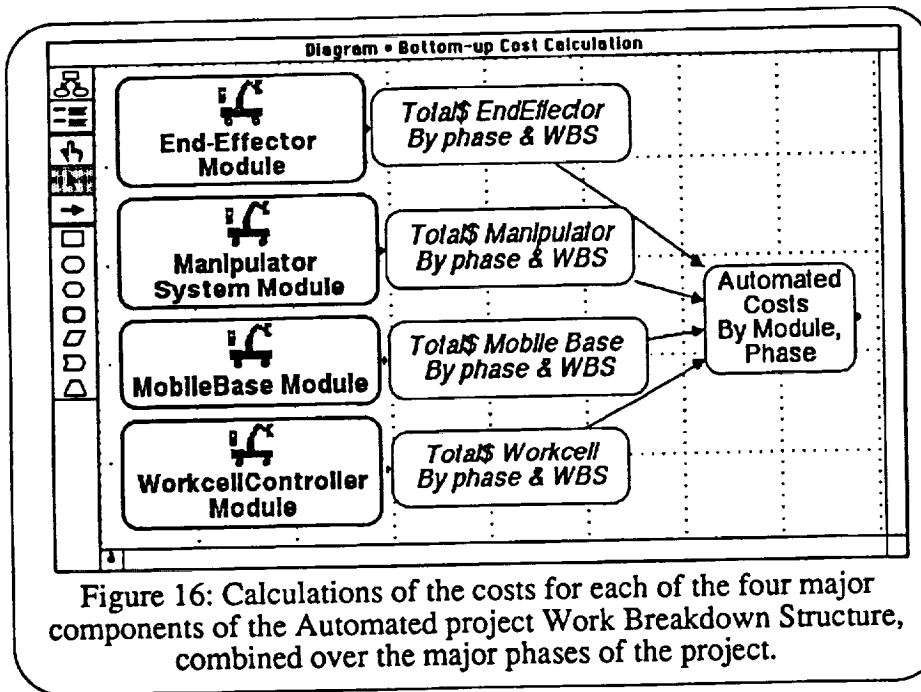


Figure 16: Calculations of the costs for each of the four major components of the Automated project Work Breakdown Structure, combined over the major phases of the project.

This submodel, accessed by double-clicking on the Bottom-Up node in the Tile WaterProofing Automated submodel, indicates the separate submodels for each of the major components of the Automation project. These components are specified in the top level Work Breakdown Structure (WBS) for the project. Within each of the component submod-

els, there may be further granularity provided for the elements of that component. The calculations for the first three components, End-Effector, Manipulator, and MobileBase, are all based on providing the bottom-up cost element values per phase as percentages of a Theoretical First Unit, i.e., proportional values based on an existing project. Since all the calculations and submodels are similar, only one will be expanded, as shown in figure 17 below. Cost values from each of the submodels are combined to form the overall bottom up cost calculations for the automated technique, as indicated in the Automated submodel (figure 15).

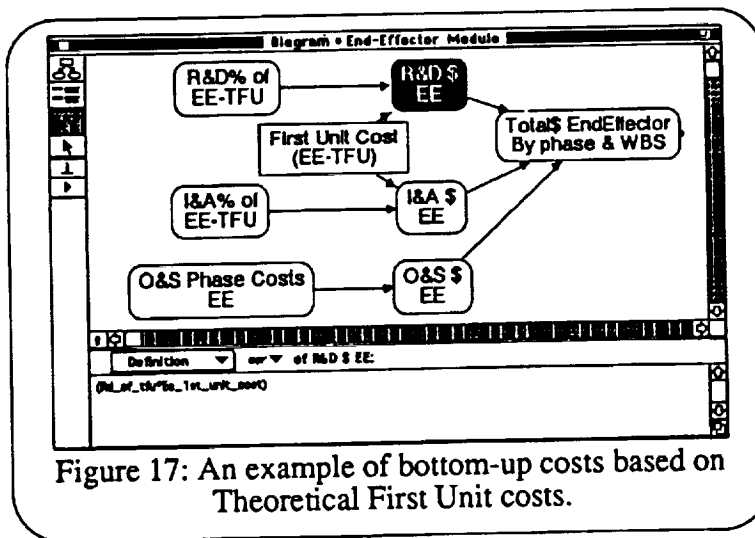


Figure 17: An example of bottom-up costs based on Theoretical First Unit costs.

In figure 17, the bottom-up costing of the End-Effector is indicated. The cost of both R&D and I&A phases, (using the same cost elements as those shown in the Manual project and the Rates submodel) are based on the modeler defining both the dollar value of a Theoretical First Unit (TFU) and the percentage for each cost element that should be used to determine the overall costs. The user may inspect both the



(Bottom-Up Cost Calculation Submodel of Automated Submodel, continued)

definition for the First Unit Cost and for the assigned percentages, as well as the definition of the conversion to dollar values; for example, the R&D\$ EE value is determined by applying the percentages assigned to the R&D cost elements to the value of the EE-TFU, using the equation

$$Rd_of_tfu * Ee_1st_unit_cost$$

This may be observed in the definition option of the attribute pane when the node R&D\$ EE is selected. The user may then navigate to the input variables to verify their definitions. The calculations for the O&S phase of the project is based on the same bottom-up technique as defined in the Manual project; the modeler enters the actual labor hours and capital costs associated with each cost element of the phase. In this submodel, all major phases are then combined into a table by summing over the cost elements per phase, and are output to the Bottom-Up submodel costs of the Automated method. The submodels for Manipulator and MobileBase are replicates of this submodel, with the appropriate percentages and TFU values entered.

The Workcell Controller Costs are for software alone. Hence, the costing for this component of the Automated method is based on the COConstructive COSt Model (COCOMO) software costing technique. When the WorkcellController submodel is expanded, the costs per phase are based on the submodel Software Development Costs, shown in figure 18.

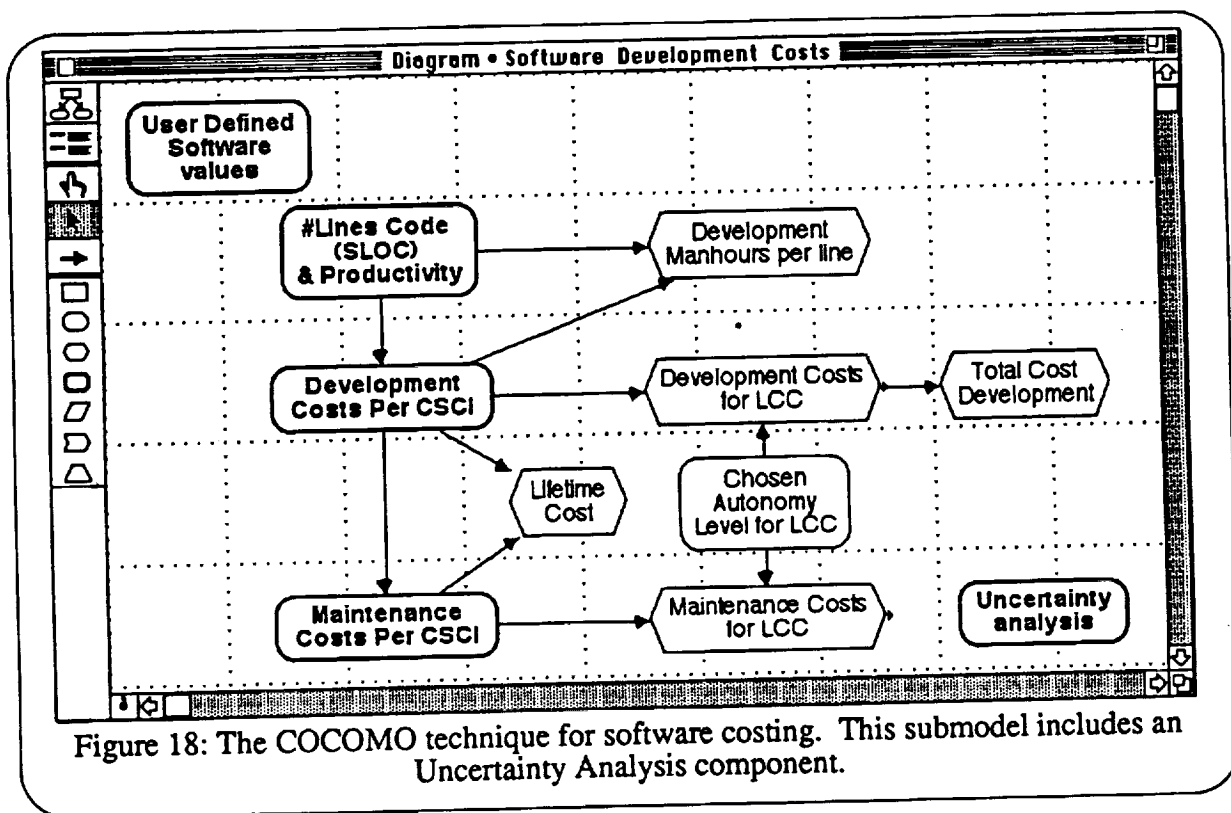


Figure 18: The COCOMO technique for software costing. This submodel includes an Uncertainty Analysis component.



(Bottom-Up Cost Calculation Submodel of Automated Submodel, continued)

As indicated, all values to be entered by the modeler are contained in the submodel **User Defined Software Values**. In this submodel, several tables are defined for including ratings for the new software project with respect to cost drivers such as required level of reliability and database size. These tables are assigned for each work-breakdown structure element for the Workcell controller; in COCOMO terminology, these elements are known as Computer Software Configuration Items, or CSCIs. In the **User Defined** submodel, the modeler also enters values specific to each CSCI such as the number of lines of source code (SLOC) of related projects, and the fraction of modifications (annual change traffic) expected. The modeler also indicates percentage relationships to the known projects and the level of uncertainty for cost drivers.

From the modeler defined parameters, the expected number of source lines of code and the nominal productivity are determined, based on existing projects (see submodel #Lines Code (SLOC) & Productivity). The overall development costs are then calculated for each CSCI, over a range of possible autonomy levels (see submodel **Development Costs Per CSCI**). Base both on modeler supplied values and the Development costs, maintenance costs are then determined for each CSCI, again over a range of possible autonomy levels (see submodel **Maintenance Costs Per CSCI**). Finally, the overall values for the software costs are defined: the development manhours per line of code and the lifetime cost per CSCI for all autonomy levels. The modeler-chosen specific autonomy level is then used to extract the development and maintenance costs that will be provided to the phase costs for the WorkCell Controller submodel.

While descriptions are included for all the elements of the COCOMO submodel, the user is referred to the text *Software Engineering Economics* by B. Boehm for further detail on the assumptions and use of the COCOMO model.

An additional aspect of the COCOMO submodel in the Demos implementation is the provision for **Uncertainty Analysis**, also known as Sensitivity Analysis. In essence, this allows the user to determine which uncertain input variables have the greatest impact on the uncertainty in a calculated variable. This analysis determines, based on both the degree of uncertainty in the input variables and the influence of its value in the calculation of the variable of interest, the variables whose uncertainty have the greatest effect on the uncertainty in that variable. This is useful in establishing the set of input variables whose range of values need to be minimized in order to provide more certainty in the variable of interest. The submodel and its output are shown in figure 19.

(Bottom-Up Cost Calculation Submodel of Automated Submodel, continued)

This Uncertainty Analysis submodel provides analysis for the effect of uncertain input variables on both the Development and Maintenance costs for the specific modeler-chosen autonomy level. The associated uncertain variables are the cost drivers for both development and maintenance calculations. The submodel itself is shown in the upper left. Output, in two formats, are also shown.

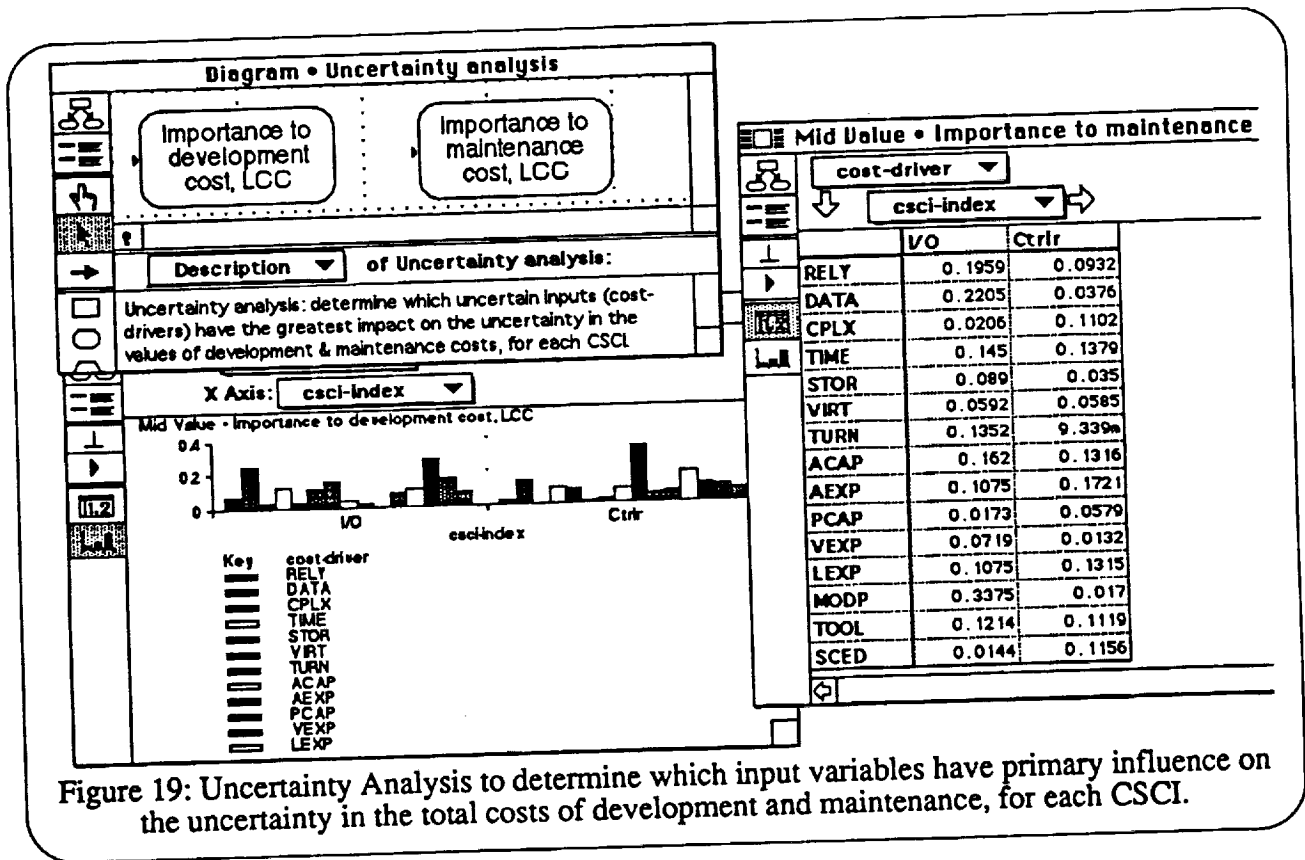


Figure 19: Uncertainty Analysis to determine which input variables have primary influence on the uncertainty in the total costs of development and maintenance, for each CSCI.

Both CSCI elements (I/O and Controller software modules) are represented in each output format. On the left, the effect of uncertain cost-driver values on the Development costs are shown graphically, indicating that uncertainty in SCED (schedule constraints) has the primary effect on uncertainty in the development costs for the I/P module, and uncertainty in DATA (database size) has the strongest effect on the development costs for the Controller software. On the right, the tabular format is shown for Maintenance, indicating that uncertainty in MODP (modern programming practices) has the most effect on uncertainty in the cost of the I/O module, while the Controller cost uncertainty is most affected by AEXP (application experience). This is most valuable to the Cost Analyst, in evaluating the effort that should be expended in determining further detail about a minimum subset of uncertain input variables.



Tile Waterproofing, Automated Submodel, Costs

Finally, on completion of cost calculation by the bottom-up methods for the Automated technique,

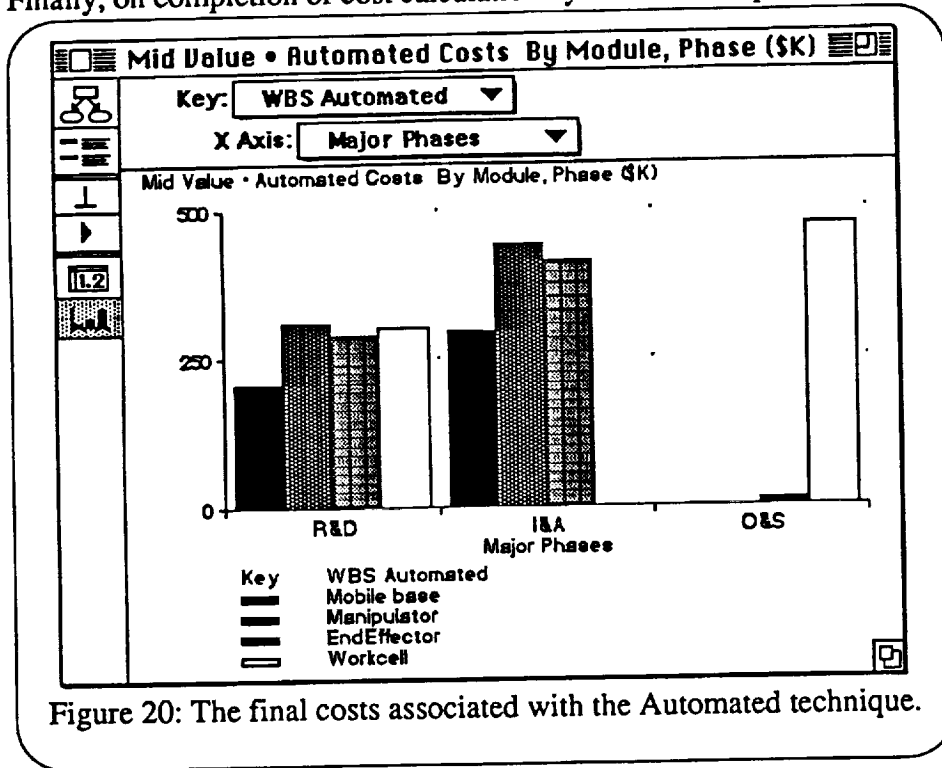


Figure 20: The final costs associated with the Automated technique.

the final costs are determined and may be displayed. As shown, they are defined for each of the major components of the Work Breakdown Structure, and are separated into the costs associated with each phase. It is these costs, summed across the WBS components, that are compared with the costs for the Manual technique as shown in figure 10 of the Cost Analysis submodel.

Comparison of a subset of these values with those generated by the top-down method is shown by displaying the Compare Bottom-Up Top-Down Costing node, as shown in figure 24.



Top-Down Cost Calculation Submodel of Automated Submodel

This submodel is a cost model to determine overall costs by using the top-down technique of Cost Estimating Relationships (CERs), based on the requirements, complexity, and technical readiness levels of key design parameters for the proposed modules, or elements of the Work Breakdown

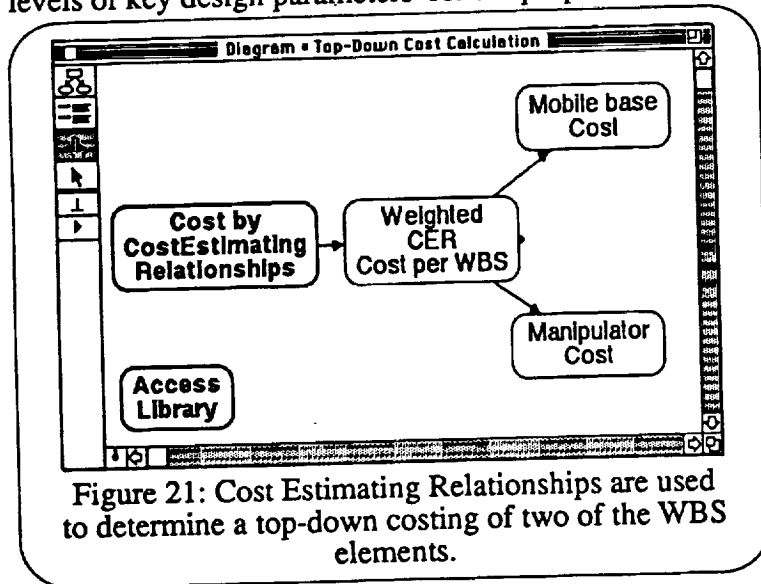


Figure 21: Cost Estimating Relationships are used to determine a top-down costing of two of the WBS elements.

Structure. Each design parameter (accuracy, payload, control type,...) has an associated complexity and technology readiness level as well as constraint values (amount of weight required, accuracy of positioning in inches,...). In this case, only the Mobile Base and the Manipulator systems are presented, to show proof of concept of providing various costing models within the framework of the ALCM.

As indicated, the submodel Cost by CostEstimating Relationships pro-

vides the cost of both the mobile base and the manipulator, for comparison to the bottom-up costing methods for the Automated technique. An access library is provided to allow easier access to table values, and will not be described further here. The submodel Cost by CostEstimating Relation-

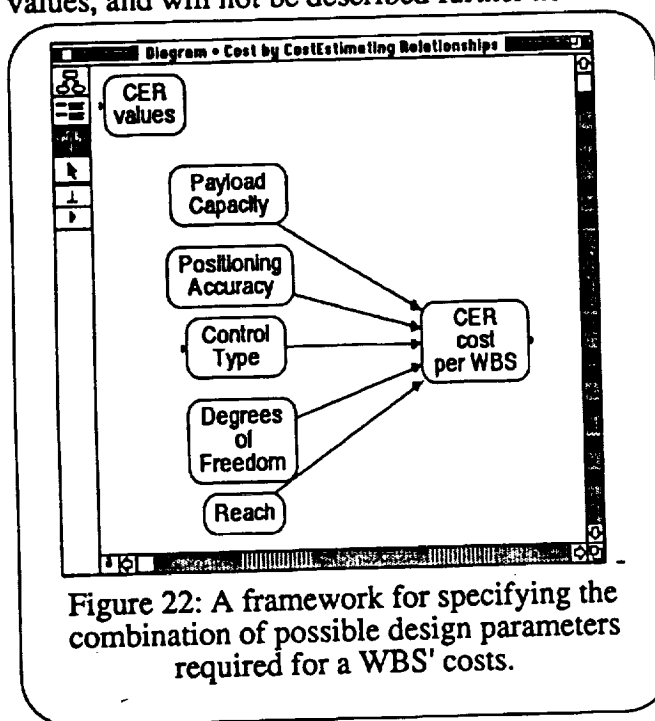
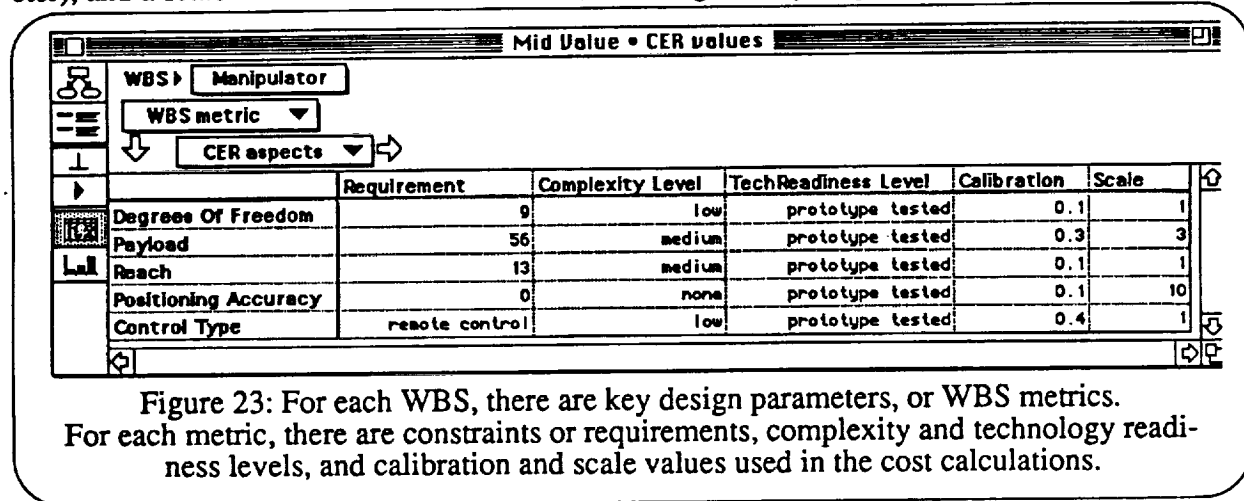


Figure 22: A framework for specifying the combination of possible design parameters required for a WBS' costs.

ships is shown in figure 22. As indicated, the key design parameters for the two modules of interest include the common parameters of Payload, Positioning Accuracy, and Control Type. Degrees of Freedom and Reach are only applicable to the Manipulator system, but are included here for the overall structure. In the table CER values, the constraint requirements, complexity levels, and technology levels are entered by the modeler. The cost of each design parameter for each WBS is based on the requirements, the complexity level and the technology readiness level.

(Top-Down Cost Calculation Submodel of Automated Submodel, continued)

Figure 23 shows the values as entered for the Manipulator system; also included are calibration constants used in the determination of the cost of requirements, for each design metric (key parameter), and a scale value used to convert units, or to neglect any inappropriate key parameters.



There is clearly more detail relating the calculation of total cost from the key parameters. The user is urged to browse through the various elements, viewing the description and optionally the definition for the various nodes.

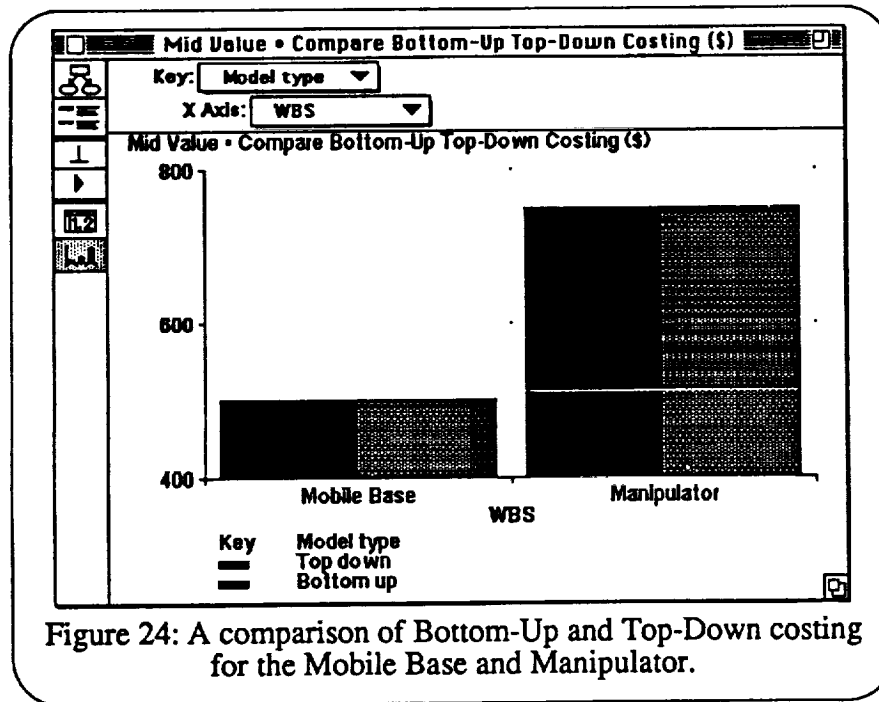


Figure 24 reverts to the Tile WaterProofing, Automated submodel, in comparing the costs determined by using both the bottom-up and top-down methods.

As indicated, the values are very close, providing the analyst with a strong level of confidence in the accuracy of both techniques. Viewing this data in a tabular format indicates a difference of approximately 10%.

Operators

39

Operators and functions often expect to work on expressions or values of a particular type. These symbols represent what type they expect:

x, y	An expression that gives a number, or an array of numbers.
u, v	An expression that gives a number or text string, or an array of numbers or text strings.
b, c	An expression that gives a boolean value true (1) or false (0), or an array of boolean values. Any non-zero number is treated as true.
a	An expression that yields an array of numbers.
i, j, k	The name of an index variable.
v	The name of a variable.
m, s, r	An expression that yields a single number (a scalar), not an array.
a	The name of an attribute.

Arithmetic operators

Operator	Meaning	Examples
+	plus	$3+2 \rightarrow 5$
-	minus	$3-2 \rightarrow 1$
*	times	$3*2 \rightarrow 3 \times 2 \rightarrow 6$
/	divided by	$3/2 \rightarrow \frac{3}{2} \rightarrow 1.5$
^	to the power of	$3^2 \rightarrow 3^2 \rightarrow 9$

Comparison operators

Operator	Meaning	Examples
<	less than	$2 < 2 \rightarrow 0$ $'A' < 'B' \rightarrow 1$
<=	less than or equal to	$2 <= 2 \rightarrow 1$ $'ab' <= 'ab' \rightarrow 1$
=	equal to	$100 = 101 \rightarrow 0$ $'AB' = 'ab' \rightarrow 0$
>=	greater than or equal to	$100 >= 1 \rightarrow 1$ $'ab' >= 'cd' \rightarrow 0$
>	greater than	$1 > 2 \rightarrow 0$ $'a' > 'A' \rightarrow 1$
<>	not equal to	$1 < > 2 \rightarrow 1$ $'A' < > 'B' \rightarrow 1$

Logical operators

Operator	Meaning	Examples
b AND c	true if both b and c are true	$1 \text{ AND } 20 < 2 \rightarrow 0$
b OR c	true if b or c or both are true, otherwise false	$0 \text{ OR } 1 < 2 \rightarrow 1$
NOT b	true if b is false, otherwise false	NOT ($2 < 3$) $\rightarrow 0$

IF b THEN x ELSE y

For values of b that are true, x is returned; for values of b that are false, y is returned.

Abs (x)

This returns the absolute value of x.

Arctan (x)

This returns the Arctangent of x in degrees. For example, remembering the venerable trigonometric identity, $\tan(x) = \sin(x)/\cos(x)$, we get:
 $\text{Arctan}(\sin(12.34)/\cos(12.34)) \rightarrow 12.34$.

Cos (x)

This returns the Cosine of x, in degrees.

Exp (x)

This returns the exponential of x, i.e., e^x .

Factorial (x)

This returns the factorial of x, which must be positive or zero.

Ln (x)

This returns the natural logarithm of x. Hence, $\ln(\exp(12.34)) \rightarrow 12.34$.

Logten (x)

This returns the log to the base 10 of x. Hence, $\text{Logten}(10^{12.34}) \rightarrow 12.34$.

Round (x)

This returns the round value of x to the nearest integer. For example, $\text{Round}(1.8) \rightarrow 2$, and $\text{Round}(1.499) \rightarrow 1$.

Sin (x)

This returns the Sine of x, x assumed in degrees.

Sqr (x)

This returns the square of x.

Sqrt (x)

This returns the square root of x, which must be positive or zero. For example, $\text{Sqr}(\text{Sqr}(12.34)) \rightarrow 12.34$.

1



You can use the **List** and **Table** options in the expression type popup to specify simple arrays and tables. Use the **Expression** option in the popup to use [] (list brackets), and **Array()** and **Table()** functions, if you need more flexibility and control in specifying arrays.

Most array functions accept an expression *a* that yields an array of numbers, and an index name *i*. The index name is optional if the array is one-dimensional. If *a* has more than one dimension, the parameter *i* should be used to specify the dimension over which to perform the function.

Area (*a*, *i*)

This computes the area under array *a* across index *i*.

Array (*i*1 , *i*2 , ... *i**n* , *y*)

This assigns a list of indices, *i*1, *i*2, ... *i**n*, as the indices of the array *y*, with *i*1 as the index of the outermost dimension, *i*2 as the second outermost, etc. *y* must have at least *n* dimensions.

Average (*a*, *i*)

This returns the mean value of an array, averaged over index *i*.

Concat (*a*, *b*, *i*, *j*, *k*)

This appends array *b* to array *a*. If they are multidimensional, then the indexes, *i* and *j* specify the dimensions of *a* and *b* respectively which are to be concatenated. If specified, *k* is the index of the resulting dimension, and will consist of the vector created by concatenating *i* and *j*.

Cumulate (*a*, *i*)

This returns an Array of the same dimensions as *a* with each element being the sum of all the elements of *a* along dimension *i* up to and including the corresponding element of *a*.

Integrate (*a*, *i*)

This applies the trapezoidal rule of integration of array *a* over index *i* and returns the result.

Max (*x*, *i*)

This returns the highest valued element of *x* (if an array) along a specified index *i*. To get the maximum of two numbers, you must make them into an array: **Max**([*a*, *b*]).

Min (*x*, *i*)

This returns the lowest valued element(s) of *x* (if an array) over a specified index *i*. To get the minimum of two numbers, you must make them into an array: **Min**([*a*, *b*]).

Normalize (*a*, *i*)

This normalizes array *a*, such that the values along index *i* sum to 1.

Product (*a*, *i*)

This returns the product of all the elements of *a*, along the dimension indexed by *i*. The resulting value has the dimensions of *a* with *i* removed.

Rank (*a*, *i*)

This returns an array of the rank values of *a* (provided that *a* is an array); the lowest value in *a* has a rank value of 1, the next-lowest has a rank value of 2, and so on.

Reform (*a*, [*i*1 , *i*2 , ... *i**n*])

This reforms a multi-dimensional array *a* in a sequence so that index *i*1 is outermost, *i*2 next outermost and so on. The indices *i*1, *i*2, etc., must be some or all of the indices of *a*.

Sequence (*r*, *s*)

This creates a one-dimensional array of successive integers from *r* to *s*. If *r* and *s* are not integers, Demos will round them first. If *s* is greater than *r*, the sequence will be increasing. If *r* is greater than *s*, the sequence will be in decreasing order.

Size (*x*)

This returns the number of elements of the outermost dimensions of an array *x*.



Slice (a, i, x)

This returns the *n*th value of array *a* over the dimension indexed by *i*. *x* must be between 1 and the length of *i*. *x* may also be an array of values, in which case, Demos will return an array of corresponding values from *a*.

Sortindex (a, i)

This computes the ranks of *a* (from smallest to largest value) and returns the items of index *i* sorted according to those ranks.

Subscript (a, i, u)

This gives the element of array *a* for which index *i* has value *u*. *u* must be one of the values of index *i*. *u* may also be an array of values from index *i*, in which case it will produce a corresponding array of resulting values from *a*. (It is essentially the same as *a[i=u]*, but it allows *a* to be a general expression, instead of restricting it to be a variable).

Sum (a, i)

This sums array *a* over the dimension indexed by Variable *i*.

Table (i1, i2, ... in) (u1, u2, u3, ... um)

This creates an *n*-dimensional array, indexed by the indices *i1, i2, ... in*. The number of indices, *n* may be 1 or more. The indices must be separated by commas and enclosed in parentheses, as shown. The second set of parameters to Table specify the values that go into the Array. These are also enclosed in parentheses, and the separating commas are optional. Each of these values is specified by an expression *u1, u2, u3, ... um*. The number of values required is the number of elements of the array, *m* which is the product of the sizes of all the dimensions. In this list of elements the last index *in* is the innermost, varying most rapidly.

Uncumulate (a, i)

This returns an array of the same dimensions as *a*. The first element (along *i*) of the result is the same as the first element of *a*. Each other element of the result is the between the corresponding elements of *a* and the previous one. It does the opposite of Cumulate. It is like a discrete differential operator.

[u1, u2, u3, ... um]

This list of expressions separated by commas and surrounded by brackets creates a one-dimensional un-indexed array, i.e., a list, whose values are *u1, u2, u3, ... um*. When a variable whose definition contains this kind of expression is computed, the computed array becomes indexed by the variable itself. Thus, index variables are typically defined in this way. This expression is also often used in the Array function expression parameter.

v [i = u]

Given *v*, a variable, and brackets enclosing an index variable name equal to an item value for that index, this returns the slice or slices of *v* along that index, like the Slice function. More than one index can be specified at a time.



Probability functions

Chancedist (*u, a, i*)

This creates a discrete probability distribution with a vector of values given in *u* and their corresponding probabilities given in a vector of probabilities *a*.

Cumdist (*a, i*)

This converts an array *a* representing cumulating probability values along Index *i* into a continuous probability distribution.

Fractiles ([*s0, s1, s2, ... sn*])

This is used to specify an arbitrary continuous distribution by a vector of *n+1* elements *si* for *i* from 0 to *n*, where *si* specifies the *i/n* fractile (quantile) of the distribution. The probability density is assumed uniform between the specified fractiles in the distribution.

Lognormal (*m, s*)

This creates a lognormal distribution with median *m* and geometric standard deviation *s*.

Normal (*m, s*)

This creates a normal probability distribution where *m* is the mean and *s* is the standard deviation.

Probdist (*a, i*)

This converts an array *a* representing probability density values along Index *i* into a continuous probability distribution.

Uniform (*r, s*)

This creates a uniform distribution between values *r* and *s*

Statistics functions

Confbands (*x*)

This returns probability or "confidence" bands over *x*, assumed to be uncertain, for probabilities specified in System variable Confidences, which by default is 5%, 25%, 50%, 75% and 95% probability.

Correlation (*x, y*)

This returns the correlation from -1 to 1 between the given distributions *x* and *y*, i.e., the degree to which the two distributions are similar, where -1 means negatively correlated, 0 means no correlation, and 1 means positively correlated.

Getfract (*x, y*)

This returns the *y*th fractile of *x*, i.e., the value which has a probability *y* of being greater than *x*. Demos evaluates *x* probabilistically.

Mean (*x*)

This returns the mean of *x* if it's a probabilistic value. Otherwise it simply returns *x*.

Mld (*x*)

This returns the mld value of an expression *x*, i.e. the value where all probabilistic inputs are replaced by their median values. Mld forces deterministic evaluation in contexts where it would otherwise be evaluated probabilistically.

Rankcorrel (*x, y*)

This computes the rank-order correlation of *x* to *y*, which is the relative strength of the distribution(s) in *x* contributing to the uncertainty distribution(s) in *y*.

Sample (*x*)

This evaluates *x* probabilistically and returns a sample of values from the distribution of *x* in an array indexed by System Variable Run.

Sdeviation (*x*)

This estimates the standard deviation of *x* from its sample if it is probabilistic. If *x* is not probabilistic, it returns 0.

Vvariance (*x*)

This returns the variance of *x* if it is probabilistic. If it isn't it returns 0. It is spelled with two Vs so that it has a different abbreviation from "Variable".

Special functions

Argmax (x, i)

This returns the corresponding value in Index *i* for which *x* is maximum.

a of v

This returns the attribute *a* of variable *v*. This is useful for adding units, titles etc. to table and graph results. Note: Demos does not automatically recompute variables that use this expression when the attribute changes.

CubicInterp (i, y, x)

This returns the natural cubic spline interpolated values of *y* along Index *i*, interpolating for values of *x*. Index *i* must be in increasing order, and must be an index of *y*. For each value of *x*, this function finds the nearest two values from *i* and uses a natural cubic spline between the corresponding values of *y* and computes the interpolated value. If *x* is below the minimum value for *i*, then the *y* value corresponding to the minimum *i* value is returned; if *x* is above the maximum value for *i*, then the *y* value corresponding to the maximum *i* value is returned.

Dydx (x, y)

This returns the derivative of expression *y* with respect to *x*, evaluated at current Midvalues. This shows how a small change in *x* affects *y*. The "small change" is 10E-6 if *x*=0, otherwise *x*/10000.

Dynamic (x1, x2, ... xn, y)

This performs dynamic simulation, used in the definition of Variables whose values change over time, and may depend on their own values at a previous time. Suppose the variable *A* is assigned the expression. The first *n* parameters are expressions giving the values of *A* for the first *n* Time periods. The last parameter *y* is an expression giving the value for each subsequent Time period, and which may refer to the Variable in earlier Time periods, e.g. it might be $A(\text{Time}-1)+Dx$.

Elasticity (y, x)

This computes the percent change in *y* caused by a 1% change in a Variable *x*. It is related to Dydx thus: $\text{Elasticity}(y, x) = \text{Dydx}(y, x) \cdot x / y$.

LinearInterp (i, y, x)

This returns linearly interpolated values of *x*, given *y* representing an arbitrary piecewise linear function. *i* is an index of input values in increasing order. *y* is an array of the corresponding output values for the function (not necessarily increasing, and may be more than one dimension). *i* must be an index of *y*. *x* may be probabilistic and/or an Array. For each value of *x*, this function finds the nearest two values from *i* and interpolates linearly between the corresponding values from *y*. If *x* is less than the first (and smallest) value in *i* it returns the first value in *y*. If *x* is greater than the last (and largest) value in *i* it returns the last value in *y*.

SubIndex (x, y, i)

This returns the index value of *i* corresponding to value *y* in Array *x*. For example, Argmax uses SubIndex(*x*, Max(*x*, *i*), *i*) to return the index value corresponding to the maximum value in *x*. If *y* is an array of values, an array of index values is returned.

Using i := x Do y

This assigns a temporary variable named *i* the value of *x* and then evaluates *y*, assumed to be an expression referring to the temporary variable *i*. *i* is essentially the same as a user-defined function parameter. You can optionally specify a parameter type qualifier to *i* by adding a colon after the temporary name *i*, followed by the qualifier name (see page xx), as, "i:p." Demos evaluates *x* according to the parameter type *p*. You can also optionally specify an index *i* to iterate over slices of *x* using the keyword In after *x*, followed by the index name, as, "In i." Each slice will be evaluated in *y*, and the results will be indexed by *i*. You can use this special syntax for simplifying complex expressions, reducing the computational effort of your model, and to be able to pass array parameters to functions that require scalar values.

WhatIf (y, x, z)

This temporarily replaces the expression *z* in the definition of variable *x* and evaluates the expression *y* (assumed to be a function of or dependant on the value of *x*), returning the result. The original definition of *x* is restored following this substitution.



A user-defined function node can be created in the diagram window in edit mode. Use the information on page 13 to create a user-defined function node, edit its title, and open its object window.

A function has one or more parameters and its definition can be an arbitrary expression containing these parameters. Parameters must be enclosed in parentheses and their names separated by commas.

By default, the expressions you pass into your function will be evaluated according to their context, i.e., deterministically or probabilistically.

Controlling the evaluation of functions

The default qualifier type is **Context**, or **Expr** (the two keywords are equivalent in Demos). To apply a different parameter type qualifier to a parameter or parameters:

- ☐ separate qualifiers from parameters by colons, e.g., (*x* : **Prob**)
- ☐ apply a qualifier to all parameters by using commas to separate the parameters and placing the qualifier after all the parameters, e.g., (*x*, *y* : **Prob**)
- ☐ apply a qualifier to only one parameter by separating the list using semi-colons, e.g., in the parameter (*x*; *y* : **Prob**), the qualifier **Prob** only applies to *y*
- ☐ separate multiple parameter-qualifier pairs by semi-colons, e.g., (*x* : **IndexT**; *y* : **Prob**).

Function parameter type qualifiers

ArrayType

Specifies that the parameter should be an Array of one or more Dimensions.

Ascending

Specifies that the parameter should be a one-dimensional list of increasing values.

Context, Expr

Default qualifier for user-defined functions. Demos evaluates Function parameters according to their surrounding context unless you apply specific qualifiers to them.

Determin

Used if the parameter should always be evaluated deterministically.

IndexType

Used if the parameter should always be an index, i.e., a list.

Numeric

Used if a parameter should be a number, or an array of numbers.

Positive

Used if a parameter should be a single positive value.

Prob

Used to evaluate the parameter probabilistically (if possible).

Sample

Used to evaluate the parameter probabilistically and checks that it is one-dimensional (i.e., one array of sample values indexed by Run).

Scalar

Used if a parameter should be a single number (scalar).

Unevaluated

Used if a parameter should not be evaluated, e.g., if it is an Array of text strings.

Vector

Used if a parameter should be a single dimension, i.e., a set of scalar numbers.



An array is a collection of numbers (or text strings) that can be treated as a single unit. Arrays are part of what makes Demos such a powerful modeling language. Operations and functions that work on single numbers generalize almost effortlessly to work on arrays. In most cases, the definition of a variable requires no change if you change the dimensions of the variables on which it depends. This makes it surprisingly easy to build models with multidimensional array values.

An array can have one or more dimensions (up to 15). A simple number (scalar) has zero dimensions. Each dimension is identified by an index, a one-dimensional array specifying its size or range of values.

You may often want to calculate a model using alternative decisions, categories (or tables) of information, a specified range of values, or alternative values for parametric analysis. A category or collection of alternative values is a dimension or index, and is a simple array. Each value in the dimension is called an item or slice. Page 19 of the Quick Reference shows you how to create a List.

	1990 1991 1992	0% 5% 10%	limit CO2 limit Methane
category: states	sequence: years	parameter: rate	decision: policy

Examples of dimensions or indices

A collection of values that correspond to items in a dimension or dimensions is table, and is the general form of an array. Page 20 of the Quick Reference shows you how to create a Table.

California	30M	limit CO2	co2_Costs
Pennsylvania	10M	limit Methane	me_Costs
Population by state		Cost function by policy	

Examples of one-dimensional tables

What happens when computing with lists and tables

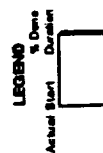
These pictures illustrate how various kinds of arrays can be combined during evaluation of variables.

	=		Identity: an index <i>i</i> is equivalent to a Table <i>i</i> indexed by itself.
	+		Operating on a List <i>i</i> and a scalar <i>r</i> produces a table <i>a</i> indexed by <i>i</i> .
	+		Operating on a table <i>a</i> indexed by <i>i</i> and a scalar <i>r</i> produces a table <i>b</i> indexed by <i>i</i> .
	+		Operating on tables <i>a</i> , <i>b</i> both indexed by <i>i</i> produces a table <i>c</i> indexed by <i>i</i> .
	+		Operating on a table <i>a</i> indexed by <i>i</i> with a table <i>d</i> indexed by <i>j</i> produces a two-dimensional result <i>e</i> for cross-product of <i>i</i> and <i>j</i> .



9.4. Programmatic

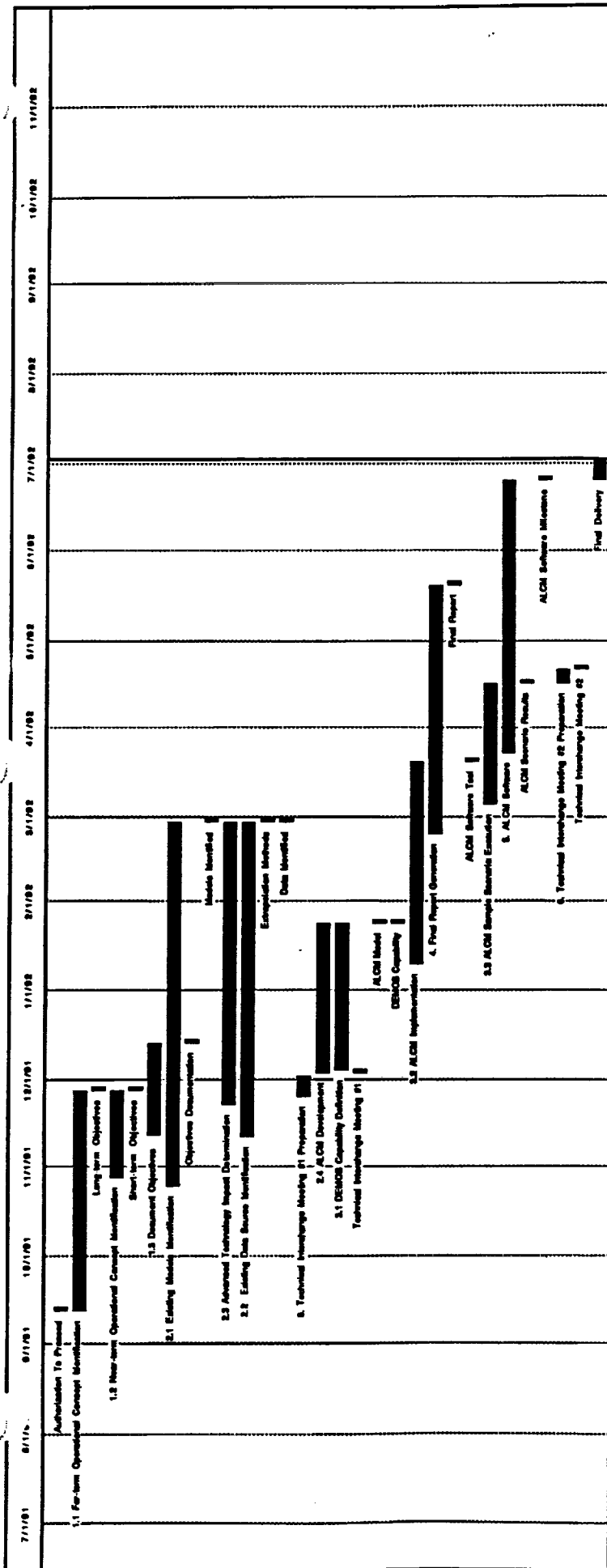






100-100000-100000





AUTOMATION LIFE CYCLE COST MODEL (ALCM)

VA-128201

ORIGINAL PAGE IS
OF POOR QUALITY



9.5. Qualitative Criteria for Robotics CER Generation

9.5.1 MOBILE BASE /PLATFORM

Cost estimation for mobile base/platform costs includes the following four criteria:

1—Payload Capability

Platforms with different categories of payload capabilities will vary in their drive system, transmission, brakes (if any), suspension, chassis/frame, material and fabrication.

1. **Drive System**—Payload weight increases require higher horsepower systems. This may require design and development of components that are not readily available such as gears, racks and pinions.
2. **Transmission**—Transmission types can vary the system cost greatly. A system could be two-wheel drive, four-wheel drive or each wheel may require an independent drive system.
3. **Brakes**—Some systems require an independent brake system to add to the system safety. This becomes more important with systems that have higher inertia (function of velocity and mass).
4. **Suspension System**—A variety of possible suspension systems exists. A suspension system could be as simple as a rocking arm or the design provides each wheel with an independent suspension system thereby minimizing the effect of one wheel on the rest of the system.
5. **Chassis/Frame**—Normally, as the payload weight increases, the chassis grows larger. This introduces problems with the design of the chaise and issues such as load distribution to minimize bending and deflection, or eliminating any possible stress points to insure long operation life.
6. **Material and Fabrication**—Higher payload capability will require use of materials that offer more strength. It is more difficult and costly to use these materials than lower strength alternatives (i.e., stainless steel 316 vs. aluminum). Furthermore, the larger the frame becomes, the more difficult it becomes to maintain dimensional tolerances and avoid frame warping. This is especially true when welding joins structural elements. In many instances, it is necessary to design and fabricate special fixtures for this purpose. Therefore, the cost of machining and fabrication increases as the payload weight increases.

2—Positioning Accuracy

Higher accuracy requires the use of more accurate positioning sensor systems. For gross positioning, a simple dead reckoning system could be sufficient. As requirements for accuracy increase, the use of sensors such as acoustic sensors, laser range finder, identification bar codes placed around the work space for triangulation, or imaging systems become necessary. Design to operate off-road may require GPS use or placement of active beacons around the operating range to allow triangulation by the system.



3—Control and Navigation

Control and navigation of the mobile base/platform can fall into the following three categories.

1. **Teleoperated/Telepresence**—This is when the operator remotely controls the movement of the platform. The operator remains in constant control and either through direct visual contact or remote sensory data it maintains its knowledge of the platform surroundings and position.
2. **Self Guided**—The platform operates along a predefined path and contains equipment to detect known obstacles. Automated Guided Vehicles (AGV) fall into this category. These systems typically operate indoors and use identification systems placed along its path.
3. **Intelligent Mobile Base/Platform**—The platform is capable of planning its path and operating on its own to reach to its commanded destination. Off-road platforms usually use this level of autonomy. The platform may use a topographic map to calculate its trajectory and it also uses its highly advanced sensor systems to avoid obstacles. The platform may use a 3-D imaging system to verify its position against the topographic map. The system becomes more sophisticated and costly when obstacles are moving objects in a 3-D space.

4—Others

The list below contains other factors that could be important in assessing system costs:

1. **Operation**—The development cost of a mobile base/platform for use in a structured environment, with known obstacles, or its movement can be predefined, is significantly less than a system designed to detect and operate around unknown obstacles.
2. **Environmental Condition**—There is a substantial increase in the cost of a system designed to operate under extreme conditions such as high temperature, high radiation or in a dusty environment.
3. **Terrain**—System design includes a variety of terrain. Cement or asphalt floors or off-road. This could also include wet or frozen surfaces.

9.5.2 MANIPULATOR SYSTEMS

There are six examination areas for estimating the cost of manipulator arms.



1—Degree of Freedom (DOF)

The increased DOF escalates the design complexity. Since it is usually desirable to avoid the design of a bulky manipulator arm, the packaging of the drive systems becomes a design challenge. This becomes more evident when there is a requirement for having a modular design to facilitate system maintenance and servicing. Furthermore, it is necessary that system singularities do not fall within its work envelope. The higher DOF translates to increased system weight, which results in payload capability reduction. Therefore, to achieve the same payload capability with increased DOF, employment of more efficient and compact motors and drive systems add to system cost.

2—Reach

Increased reach introduces new design issues including: increase in the system weight and the difficulty in controlling flexible modes to maintain a high level of control. Resolving these issues raise the cost significantly. The solution includes the use of more expensive materials that offer high stiffness-to-weight ratios with more advanced control algorithms.

3—Payload Capability

Higher payload capability results from the use of motors with higher horsepower or by reducing manipulator arm linkages and drive system weights. Both of these two options translate directly to higher cost.

4—Positioning Accuracy and Repeatability

Higher accuracy is a function of mechanical design, sensor systems and control. The next section discusses the latter. Higher positioning accuracy requires tighter design tolerances where there is no free-play between the elements. This includes, the drive system and all the associated gears and tape drive systems (if any). To avoid the cost of fabricating and assembling high precision hardware with a long operating life, a design may use a direct drive at the joints. This will have its own drawbacks as discussed before (packaging, motor size, etc.). Use of a direct drive could help to reduce system hysteresis and/or uncontrolled compliance. The use of more advanced sensors with higher resolution and update rates is another factor that increases system accuracy along with increasing cost.

5—Command & Control

There is a wide variety of control architectures and algorithms used to control manipulator systems. They vary from a simple PID controller to an adaptive and model reference control system capable of optimizing system performance.

Additionally, the interface between man/operator and the machine could also vary drastically depending on the requirements. For the most part they are very similar to those of the mobile base (discussed above). The command and control categories contain the following five considerations:



1. **Preprogrammed**—This is the most primitive level of interaction between man and robotic manipulator arms. Training permits the manipulator arm to perform a task through a series of motions. When the robot is in operation mode, it performs the same task repeatedly. This is a very common technique used in assembly line manipulator arms. After training, the robot does not interact with the operator and has limited interaction with the surroundings.
2. **Teleoperated/Telepresence**—Here, a master arm or a joystick controls the manipulator system, referred to as a slave arm. Usually there is a one-to-one correlation between the master arm and slave arm degrees of freedom. The position and orientation of each master arm joint are detected, multiplied by a factor (gain) and sent as a command to the slave arm. Some of the more complicated arms may also detect the force at the slave and feed a portion of that back to the master arm and the operator (bilateral force feedback). In the case of a joystick and/or a track-ball, the operator flies the end effector. To command the manipulator arm, an inverse kinematic algorithm calculates each joint position and orientation based on the commanded position.
3. **Supervisory Control**—Here the manipulator arm is capable of decomposing low level tasks into subtasks. The arm then performs these subtasks in an orderly manner. The operator interface with the robot will be through higher level commands and the operator will supervise the operation with manual control and emergency shut down capabilities.
4. **Coordinated Motion**—This involves using more than one manipulator to perform tasks. Coordination of manipulator arm motions avoids damage to the work piece and/or the arms. For example, when a manipulator arm pulls, the other(s) will give in and vice versa. The controller strives to minimize the forces and torques in one or more axis depending on the task performed and the controller requirements. This also requires the use of sensors to detect the forces and torques in each axis.
5. **Task Based Control**—This is the highest level of operator interface. The robot workcell controller receives task objectives. The workcell controller decomposes these objectives into tasks, subtasks and motions and executes them accordingly based on available resources (various manipulator arms, sensors, etc.). In the event a problem arises during task execution, the controller can develop a contingent set of subtasks and execute them to work around the problem(s).

6—Others

There are other factors such as high velocity requirements or harsh operating environments that could impact the system cost. For example, many welding robots require their electronics and all the power and data lines shielding to maintain signal integrity and



protect them from noise induced by welding apparatus. The cost impact of these types of system requirements warrants individual consideration and not explicitly included as part of the model. However, the means of expressing these factors in the model should be present.

9.5.3 SPECIAL TOOLS (END EFFECTOR)

End effectors perform a wide range of functions. Depending on the function, they vary in the design and cost. In many cases it is the end effector that makes one robot uniquely different from others. The same manipulator arm can perform completely different functions with different types of end effectors. On the other hand, the robot performs the same function with a different work envelope using the same end effector and manipulator replacement. Because of the uniqueness of the end effector design requirements, it is very difficult to include them in a generalized cost model. We recommend the initial use of the following six criteria for estimating end effector cost:

1—Handling

End effectors generally perform a physical function on the work-piece (contact) or monitor/inspection (non-contact). They vary in accuracy and load handling capability. Usually, the end effector cost increases as the accuracy and load carrying capability increases.

1. **Contact End Effector**—The end effector encounters the work-piece and performs a function or grasps an object while the manipulator arm moves it from one point to another. Use of the most basic end effector, the parallel jaw, includes a variety of applications including PC-board assembly and manipulation of objects in assembly lines. Common use also includes power tools, such as a screw driver. The cost increases as the tool becomes more specialized.
2. **Non-contact End Effector**—Their use includes movement around a sensor system package to perform non-destructive testing (NDT) or monitoring an on-going operation. Therefore, they carry a fixed mass. Inspection tasks may use a more complicated system to guide a small probe into a maze-like system. This may require more system flexibility and degrees of freedom.

2—Environment

The operating environment impacts the end effector more than the rest of the robot. In many cases, the end effector is the main element in creating an adverse environment. For example in the case of the re-waterproofing tool, the end effector injects DMES, which is not only hazardous but reacts to most materials and causes corrosion. A welding end effector is another example of an end effector type that falls into this category. The operating environment itself is another factor. Some operating environments impose additional requirements. For example, a nuclear hot cell uses a hardened end effector and an underwater application uses a water resistance or sealed end effector.

3—Force/Torque and Micro Positioning Control

Most manipulator arm overall control schemes include end effector control systems. However, there are those which require their own closed-loop control system. These are the end effectors that offer multiple DOFs. Operations include these end effectors handling delicate objects or as micro positioning systems (the re-waterproofing end effector serves both these purposes). In each case, the end effector design is more complicated and uses more sensors to control forces, torques and positioning accuracy.

4—Machine Vision

Vision systems can be independent of the end effector. However, some applications, such as monitoring, use vision as part of the end effector. In either case, it's an important part of robotics with wide use by industry. Most currently used vision systems are off-the-shelf and operate under a specific environment using a set of standard algorithms. Vision system attributes include easy integration and modest cost. The cost of vision systems increases for unstructured environments. The following applications may use vision systems:

1. **Inspection and monitoring**—Vision system uses include unfamiliar environment location and identification of specific features under varying lighting and orientation. It may also perform with partially obscured objects. This level of vision system capability may not be available off-the-shelf and could require a sizable development cost.
2. **Guiding a robot (path planning and collision avoidance)**—Along with the capabilities outlined above, vision system capabilities include use of its images to generate a map of its surroundings. 3-D image generation may require a stereo vision system. Trajectory determination for the robot movement uses this map. This becomes more complicated when the vision system operates in hostile outdoor terrain. Development of this capability for an operational system will be costly and will require large R&D investment.

5—Laser system

Typical laser usage includes being a range finder or scanner. Many applications have proven them accurate and reliable. However, if it is a high power laser beam, it could be harmful and may require an expensive and elaborate set of safe guards. Part of the laser system cost includes the safe guards.

6—Others

There are many other criteria for consideration in cost estimating the end effectors/special tools. These include each end effector's peculiarities. For example, an end effector positions an X-ray imaging device. Unique issues require separate consideration.



•



9.6. Robotics DDT&E CER Tables

Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Mobile Base/Platform					500
Payload Capability (lb) (PL)					
	PL>250	(L,M,H)	6		
	250<PL<1000		6		
	1000<PL<10000		6	H	
	10000<PL		6		
	Others		—		
Positioning Accuracy (IN) (PA)					
	PA>1.0	(L,M,H)	6		
	1.0>PA>0.1		6		
	0.1>PA		6	M	
	Others		—		
Control & Navigation (CN)					
	Remote control	(L,M,H)	6		
	Self-guided (AGV type)		6		
	Path planning capability		4	L	
	Others		—		



Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Manipulator Systems					750
Degree of Freedom (DOF)					
	DOF ≤ 6	(L,M,H)	6	L	
	DOF > 6		6		
	Others		—		
Reach (FT) (R)					
	10 > R	(L,M,H)	6	M	
	10 < R < 50		6		
	Others		—		
Payload Capability (lb) (PC)					
	10 > PC	(L,M,H)	6	M	
	10 < PC < 25		6		
	25 < PC < 100		6		
	Others		—		
Positioning Accuracy (in) (PA)					
	0.1 < PA	N/A	6		
	0.001 < PA < 0.1		6		
	Others		—		
Command & Control (CC)					
	PID/	(L,M,H)	6	L	
	Conventional		6		
	Modern		6		
	Control		4		
	Integrated		4		
	Advanced		4		
	Sensors		4		
	Others		—		



Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Special Tools (Re-waterproofing End Effector)					700
Handling	Contact Tool	(L,M,H)	5	H	
	Non-contact/		5		
	Inspection				
	Others		—		
Environment (Toxicity)					
	None	(L,M,H)	—		
	Toxic		—	H	
Force/torque & Micro-positioning Control					
	None	(L,M,H)	—	—	
	Low/Medium		6		
	Resolution				
	High		5	M	
	Resolution				
Machine Vision					
	None		—		
	Low/Medium	(L,M,H)	6		
	Resolution				
	High		5		
	Resolution				
Laser System					
	None		—		
	Low Power	(L,M,H)	6		
	(safe)				
	High Power		6		
	(unsafe)				



Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Workcell Controller					300
Autonomy Level					
	None	—			
	Little	(L,M,H)	6	L	
	High	(L,M,H)	4		
Communication Protocol					
	Standard	(L,M,H)	6	L	
	Customized	(L,M,H)	5		
System Integration					200
Mechanical					
	—	(L,M,H)	—	L	
Electronics					
	—	(L,M,H)	—	L	
Certification					300
Ground Support Equipment					
	—				
Maintenance Facility					500
Electronics					
	—	(L,M,H)	—	M	
Mechanical					
	—	(L,M,H)	—	L	
Laser					
	—	(L,M,H)	—	L	
Optics					
	—	(L,M,H)	—	L	
Toxic					
	—	(L,M,H)	—	M	
Others					
	—	—	—	—	
KSC Project Management					400
	—	(L,M,H)	—	M	



Additional Data Points

The cost data are from off-the-shelf hardware that could be used for model construction.

Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Mobile Base/Platform					40-100
Payload Capability (lb) (PL)					
	PL>250	(L,M,H)	6	L	
	250<PL<1000		6		
	1000<PL<10000		6		
	10000<PL		6		
	Others		—		
Positioning Accuracy (IN) (PA)					
	PA>1.0	(L,M,H)	6	M	
	1.0>PA>0.1		6		
	0.1>PA		6		
	Others		—		
Control & Navigation (CN)					
	Remote control	(L,M,H)	6	M	
	Self-guided (AGV type)		6		
	Path planning capability		4		
	Others		—		

Off-the-shelf AGV



Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Manipulator Systems					60-120
Degree of Freedom (DOF)					
	DOF≤6	(L,M,H)	6	M	
	DOF>6		6		
	Others		—		
Reach (FT) (R)					
	10>R	(L,M,H)	6	L	
	10<R<50		6		
	Others		—		
Payload Capability (lb) (PC)					
	10>PC	(L,M,H)	6		
	10<PC<25		6	M	
	25<PC<100		6		
	Others		—		
Positioning Accuracy (in) (PA)					
	0.1<PA	N/A	6		
	0.001<PA<0.1		6	X	
	Others		—		
Command & Control (CC)					
	PID/	(L,M,H)	6	M	
	Conventional				
	Modern		6		
	Control				
	Integrated		4		
	Advanced				
	Sensors				
	Others		—		

Off-the-shelf manipulator arm



Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Special Tools (Parallel Jaw)					5-15
Handling	Contact Tool	(L,M,H)	5	L	
	Non-contact/ Inspection		5		
	Others		—		
	Environment (Toxicity)				
	None	(L,M,H)	—	—	
	Toxic		—		
Force/torque & Micro-positioning Control					
	None	(L,M,H)	—	—	
	Low/Medium Resolution		6		
	High Resolution		5		
Machine Vision					
	None		—	—	
	Low/Medium Resolution	(L,M,H)	6		
	High Resolution		5		
Laser System					
	None		—	—	
	Low Power (safe)	(L,M,H)	6		
	High Power (unsafe)		6		

Off-the-shelf parallel jaw



Hardware System	Capability Range	Complexity Level (CL)	Technology Readiness (TL)	KSC Robot CL	Parameter Cost (\$K)
Special Tools (Welding End Effector)					25-50
Handling	Contact Tool	(L,M,H)	5	M	
	Non-contact/		5		
	Inspection				
	Others		—		
Environment (EMI, Temperature)					
	None	(L,M,H)	—	—	
	EMI	(L,M,H)	6	M	
	Temperature	(L,M,H)	6	M	
Force/torque & Micro-positioning Control					
	None	(L,M,H)	—	—	
	Low/Medium		6		
	Resolution				
	High		5		
	Resolution				
Machine Vision					
	None		—	—	
	Low/Medium	(L,M,H)	6		
	Resolution				
	High		5		
	Resolution				
Laser System					
	None		—	—	
	Low Power	(L,M,H)	6		
	(safe)				
	High Power		6		
	(unsafe)				

Off-the-shelf welding end effector



REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 5/22/92		3. REPORT TYPE AND DATES COVERED Draft Final Report (9/3/91 - 5/1/92)
4. TITLE AND SUBTITLE Automation Life-Cycle Cost Model Final Report			5. FUNDING NUMBERS NAS2-13419	
6. AUTHOR(S) Thomas P. Gathmann Arlinda J. Reeves Rick Cline Dr. Max Henrion Corinne Ruokangas				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Rockwell International Space Systems Division 12214 Lakewood Blvd. Downey, CA 90241			8. PERFORMING ORGANIZATION REPORT NUMBER SSD92D0154	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Henry Lum FI: 244-7 NASA-AMES Research Center Moffett Field, CA 94035			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified - Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The problem domain being addressed by this contractual effort can be summarized by the following list: - o Automation and Robotics (A&R) technologies appear to be viable alternatives to current, manual operations - o Life-cycle cost models are typically judged with suspicion due to implicit assumptions and little associated documentation - o Uncertainty is a reality for increasingly complex problems and few models explicitly account for its affect on the solution space The objectives for this effort range from the near-term (1-2 years) to far-term 3-5 years). In the near-term, the envisioned capabilities of the modelling tool are annotated. In addition, a framework is defined and developed in the Decision Modelling System (DEMOS) environment. Our approach is summarized as follows: - o Assess desirable capabilities (structure into near- and far-term) - o Identify useful existing models/data - o Identify parameters for utility analysis - o Define tool framework - o Encode scenario thread for model validation - o Provide transition path for tool development This report contains all relevant, technical progress made on this contractual effort.				
14. SUBJECT TERMS Automation Robotics Life-Cycle Cost Modelling			15. NUMBER OF PAGES 169	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified
			20. LIMITATION OF ABSTRACT Unlimited	



